AIMS Mathematics, 11(8): 24194–24224.
DOI: 10.3934/math.2026977
Received: 25 May 2026
Revised: 16 July 2026
Accepted: 24 July 2026
Published: 07 August 2026

https://www.aimspress.com/journal/Math

Research article

A subspace minimization Riemannian conjugate gradient method with nonmonotone Wolfe line search and a restart strategy

Shajie Xing[1], **Huangyue Chen**[2] **and Chunming Tang**[2,*]

[1] School of Physical Science and Technology, Guangxi University, Nanning 530004, China

[2] School of Mathematics & Center for Applied Mathematics of Guangxi, Guangxi University, Nanning 530004, China

* **Correspondence:** Email: cmtang@gxu.edu.cn.

Abstract: This paper proposes a subspace minimization Riemannian conjugate gradient (CG) method combined with nonmonotone Wolfe line search and a restart strategy. First, an improved nonmonotone version of the Riemannian Wolfe line search is proposed to address a numerical drawback of the Riemannian Wolfe line search. Second, the search direction is determined by minimizing a quadratic approximation model of the objective function in a two-dimensional subspace, with the CG parameters dynamically adjusted via four decision criteria based on the local curvature information and the inner product between the current gradient and the previous search direction. This direction satisfies the sufficient descent condition. In addition, to accelerate convergence, a quantity measuring the degree of local quadratic approximation of the objective function is introduced as a restart criterion, thereby establishing the complete framework of the proposed method. Under mild conditions, the method is proven to achieve global convergence and R-linear convergence. Finally, comparative experiments on seven test problems demonstrate that the proposed method outperforms several existing competitive approaches in terms of the number of iterations and CPU time.

Keywords: conjugate gradient method; Riemannian optimization; subspace technique; nonmonotone Wolfe line search; restart strategy
Mathematics Subject Classification: 65K05, 90C30

1. Introduction

In this paper, we address the following Riemannian optimization problem:

$$\min f(x), \quad \text{s.t. } x \in \mathcal{M}, \tag{1.1}$$

where $\mathcal{M}$ is a Riemannian manifold and $f\colon \mathcal{M} \to \mathbb{R}$ is a continuously differentiable objective function. Problem (1.1) finds broad applications in numerous fields, including the linear eigenvalue problem [1, 2], low-rank matrix and tensor completion [3, 4], joint diagonalization problems [5], signal processing [6], as well as the orthogonal procrustes problem [7, 8]. Unlike Euclidean optimization, solving problem (1.1) requires specialized tools including retraction and vector transport (defined in the next section) to address the nonlinear geometric structure of the manifold.

In the Euclidean case, i.e., when $\mathcal{M} = \mathbb{R}^n$ in (1.1), the conjugate gradient (CG) method was originally proposed by Hestenes and Stiefel [9] in 1952 for solving linear systems with a symmetric positive definite coefficient matrix. Subsequently, Fletcher and Reeves [10] extended the CG method to unconstrained nonlinear optimization problems. This theoretical breakthrough spurred extensive research on nonlinear CG methods. Polak and Ribiere [11] and Polyak [12] extended the CG framework to nonlinear problems. Dai and Yuan [13, 14] proposed efficient variants with global convergence. Zhang et al. [15] developed a descent-type method. Wan et al. [16] and Xiao et al. [17] introduced modified update strategies. Liu et al. [18] and Jiang et al. [19] proposed spectral and three-term variants, respectively. To improve the computational efficiency of CG methods for large-scale optimization problems, Yuan and Stoer [20] first introduced a subspace minimization CG (SCG) algorithm, which determines the search direction by minimizing a quadratic approximation model of the function f over the two-dimensional subspace spanned by the current gradient vector and the latest search direction. Andrei [21] developed a three-dimensional SCG method. Furthermore, various improved variants have been extensively investigated. For instance, Dai and Kou [22] incorporated the Barzilai-Borwein (BB) idea [23] into the SCG method, resulting in a class of BB CG methods. Li et al. [24] analyzed a novel SCG algorithm with nonmonotone Wolfe line search, and Liu and Liu [25] further improved this algorithm. Liu et al. [26] proposed a new SCG method based on the projection technique used in memoryless quasi-Newton methods.

The extension of CG methods from Euclidean space to Riemannian manifolds has become a vibrant research field. Early Riemannian CG methods, e.g., those by Lichnerowicz [27] and Smith [28], relied on exponential mapping and parallel transport. However, subsequent work by Absil et al. [29] revealed that these geometric operations could be computationally expensive, leading to the introduction of retraction and vector transport as more efficient alternatives. Based on this theoretical advancement, researchers generalized classical CG methods to Riemannian manifolds, including the Fletcher-Reeves (FR), Dai-Yuan (DY), Hestenes-Stiefel (HS), and Polak-Ribière-Polyak methods. These developments established a Riemannian CG framework with provable convergence properties [30–32]. Further studies extended this framework through various hybrid strategies [33–35]. More recently, additional hybrid variants have been investigated [36]. Riemannian spectral CG methods have also attracted considerable attention [37–39]. Several other spectral CG variants were proposed to enhance the practical performance of Riemannian optimization algorithms [40, 41]. In addition, three-term Riemannian CG methods have been developed to improve convergence behavior [42, 43].

1.1. Contributions

Compared to the mature theory of Euclidean SCG methods, most current Riemannian CG methods rely on monotonic line searches and lack local convergence rate guarantees. To overcome these limitations and enhance scalability, this paper presents a subspace minimization Riemannian CG

method equipped with nonmonotone Wolfe line search and a restart strategy. The key contributions are listed below:

(i) We propose a novel Riemannian CG method that integrates subspace techniques, nonmonotone Wolfe line search, and a restart strategy. A nonmonotone version is proposed to address the numerical drawback of the Riemannian Wolfe line search, namely the tendency to produce overly conservative stepsizes and the possible failure of the Armijo condition due to round-off errors or ill-conditioning, which can significantly slow down convergence. The search direction, which satisfies the sufficient descent condition, is obtained by minimizing a quadratic approximation model of f within the two-dimensional subspace spanned by the current gradient and the previous search direction, with the CG parameters dynamically adjusted via four decision criteria.

(ii) To accelerate convergence, a quantity is introduced to evaluate the degree of the local quadratic approximation of the function f, which is then employed as the restart criterion. A complete algorithmic framework is established. It is particularly noteworthy that the proposed method guarantees global convergence without requiring the vector transport to satisfy the non-expansive condition [44]. Furthermore, under the assumption that the function f is uniformly retraction-convex, the algorithm is proven to achieve R-linear convergence.

(iii) The performance of the proposed method is analyzed through numerical experiments conducted on seven test problems with varying dimensions. The proposed method is systematically compared with eleven existing approaches, including two classical Riemannian BB methods, six Riemannian CG methods under two line search conditions, and three Riemannian hybrid CG methods under two line search strategies. Experimental results show that the proposed method achieves faster convergence speed and stronger stability, exhibiting excellent numerical performance during iterations.

1.2. Organization and preliminaries

The rest of this paper is structured as follows. Section 2 introduces the nonmonotone Riemannian Wolfe conditions and establishes a general algorithmic framework. Section 3 proposes a subspace minimization Riemannian CG method with nonmonotone Wolfe line search and a restart strategy, and establishes its global convergence. Section 4 proves the R-linear convergence rate. Section 5 presents numerical experiments. Section 6 concludes the paper.

We briefly review the core notation and concepts of Riemannian optimization [29, 45, 46]. Given a d-dimensional Riemannian manifold $\mathcal{M}$, let $T_x\mathcal{M}$ denote the tangent space at

$$x \in \mathcal{M} \text{ and } T\mathcal{M} = \bigcup_{x\in\mathcal{M}} T_x\mathcal{M}$$

denote the tangent bundle. $\langle\cdot,\cdot\rangle_x$ denotes the inner product on $T_x\mathcal{M}$ that induces the norm

$$\|\xi\|_x := \sqrt{\langle\xi,\xi\rangle_x}$$

for a tangent vector $\xi \in T_x\mathcal{M}$. The flat of a tangent vector $a \in T_x\mathcal{M}$ is denoted by $a^\flat$, i.e., $a^\flat\colon\ T_x\mathcal{M} \to \mathbb{R},\ v \mapsto \langle a, v\rangle_x$. The Riemannian gradient $\mathrm{grad} f(x) \in T_x\mathcal{M}$ of f at x is the unique tangent vector

satisfying $\mathrm{D}f(x)[v] = \langle v, \mathrm{grad}f(x)\rangle_x$ for all $v \in T_x\mathcal{M}$, where $\mathrm{D}f(x)$: $T_x\mathcal{M} \to \mathbb{R}$ is the differential of f at x. In addition, the Riemannian Hessian $\mathrm{Hess}f(x)$: $T_x\mathcal{M} \to T_x\mathcal{M}$ of f is a linear operator that acts on a tangent vector $\xi \in T_x\mathcal{M}$ as $\mathrm{Hess}f(x)$: $\xi \mapsto \nabla_\xi \mathrm{grad}f$, where ∇ is the Riemannian (Levi-Civita) connection [29]. A retraction on a manifold $\mathcal{M}$ is a smooth map R: $T\mathcal{M} \to \mathcal{M}$ whose restriction R_x: $T_x\mathcal{M} \to \mathcal{M}$ at each point $x \in \mathcal{M}$ satisfies: (i) $R_x(0_x) = x$, where 0_x denotes the origin of $T_x\mathcal{M}$; (ii) $\mathrm{D}R_x(0_x) = \mathrm{id}_{T_x\mathcal{M}}$, where $\mathrm{id}_{T_x\mathcal{M}}$ is the identity map on $T_x\mathcal{M}$. A vector transport on a Riemannian manifold $\mathcal{M}$ is a smooth map $\mathcal{T}$: $T\mathcal{M} \oplus T\mathcal{M} \to T\mathcal{M}$: $(\eta, \xi) \mapsto \mathcal{T}_\eta(\xi)$ (where $\oplus$ denotes the Whitney sum, i.e., $T\mathcal{M} \oplus T\mathcal{M} := \{(\eta, \xi) : \eta, \xi \in T_x\mathcal{M},\ x \in \mathcal{M}\}$) satisfying

(i) $\mathcal{T}_\eta(\xi) \in T_{R_x(\eta)}\mathcal{M}$ for all $\eta, \xi \in T_x\mathcal{M}$;

(ii) $\mathcal{T}_{0_x}(\xi) = \xi$ for all $\xi \in T_x\mathcal{M}$;

(iii) $\mathcal{T}_\eta(a\xi + b\zeta) = a\mathcal{T}_\eta(\xi) + b\mathcal{T}_\eta(\zeta)$ for all $a,\ b \in \mathbb{R}$ and $\eta, \xi, \zeta \in T_x\mathcal{M}$.

This paper employs the differentiated retraction as the vector transport, defined by $\mathcal{T}_\eta(\xi) = \mathrm{D}R_x(\eta)[\xi]$, $\eta, \xi \in T_x\mathcal{M}$.

2. Nonmonotone Riemannian wolfe line search

This section proposes a nonmonotone Riemannian Wolfe line search, develops a general framework, and establishes its global convergence under mild conditions.

Using the retraction, the iterative scheme for problem (1.1) is given by

$$x_{k+1} = R_{x_k}(\alpha_k \eta_k), \tag{2.1}$$

where $\alpha_k > 0$ is the stepsize and $\eta_k \in T_{x_k}\mathcal{M}$ is the search direction. Let $\{x_k\} \subset \mathcal{M}$ be a sequence defined by (2.1). To simplify notation, we denote $g_k := \mathrm{grad}f(x_k)$ and $g_x := \mathrm{grad}f(x)$. This notation will be adopted hereafter if no confusion occurs. The Riemannian Wolfe conditions are summarized as follows. Given a point x_k and a search direction η_k, the stepsize α_k is required to satisfy

$$f(R_{x_k}(\alpha_k \eta_k)) \le f(x_k) + \sigma_1 \alpha_k \langle g_k, \eta_k\rangle_{x_k}, \tag{2.2}$$

$$\langle g_{k+1}, \mathrm{D}R_{x_k}(\alpha_k \eta_k)[\eta_k]\rangle_{R_{x_k}(\alpha_k\eta_k)} \ge \sigma_2 \langle g_k, \eta_k\rangle_{x_k}, \tag{2.3}$$

where $0 < \sigma_1 < \sigma_2 < 1$. The Riemannian Wolfe conditions (2.2) and (2.3) serve as essential components in establishing convergence guarantees for gradient-type optimization methods [30–34, 37, 44].

In practical numerical computations, due to numerical issues such as round-off errors or ill-conditioning, condition (2.2) may fail to be satisfied, a phenomenon analogous to that observed in Euclidean space (see [47, Section 3]). Moreover, such monotonicity constraints tend to yield overly conservative stepsizes, which can degrade the convergence and efficiency of the algorithm. Therefore, inspired by the nonmonotone strategies in the Euclidean space [25, 47], we introduce

$$f(R_{x_k}(\alpha_k \eta_k)) \le f(x_k) + \delta_k + \sigma_1 \alpha_k \langle g_k, \eta_k\rangle_{x_k}, \tag{2.4}$$

where $\delta_k \ge 0$ satisfies $\lim\limits_{k\to+\infty} k\delta_k = 0$. The line search satisfying conditions (2.3) and (2.4) is referred to as the *nonmonotone Riemannian Wolfe line search*. Since any α_k satisfying (2.2) and (2.3) necessarily

satisfies (2.3) and (2.4), and such an α_k exists by [44], there exists a stepsize α_k that simultaneously meets conditions (2.3) and (2.4). According to [48], we select δ_k to ensure R-linear convergence of the proposed method as

$$\delta_k = \begin{cases} 0, & \text{if } k = 0, \\ \min\left\{\frac{1}{k\log\left(\frac{k}{d}+12\right)}, C_k - f(x_k)\right\}, & \text{if } k \geq 1, \end{cases} \tag{2.5}$$

where d denotes the dimension of $\mathcal{M}$ (see Section 1.2), C_k is defined by the convex combination of $f(x_k)$ and C_{k-1} as

$$C_k = \frac{t_{k-1}Q_{k-1}C_{k-1} + f(x_k)}{Q_k}, \tag{2.6}$$

with $Q_k = t_{k-1}Q_{k-1} + 1$, $0 \leq t_{\min} \leq t_k \leq t_{\max} < 1$, initialized with $Q_0 = 1$ and $C_0 = f(x_0)$. Based on the nonmonotone Riemannian Wolfe line search, we now describe the algorithmic framework in Algorithm 1.

Algorithm 1 A generic algorithmic framework.

Step 0: Choose an initial iterate $x_0 \in \mathcal{M}$ and a tolerance $\varepsilon \geq 0$. Set $k := 0$ and $\eta_0 = -g_0$.

Step 1: If $\|g_k\|_{x_k} \leq \varepsilon$, then **stop**.

Step 2: Determine α_k satisfying the nonmonotone Riemannian Wolfe line search (2.3) and (2.4).

Step 3: Set $x_{k+1} = R_{x_k}(\alpha_k\eta_k)$ and compute $f(x_{k+1})$ and g_{k+1}.

Step 4: Calculate the search direction η_{k+1}.

Step 5: Set $k := k + 1$ and go to Step 1.

The global convergence analysis of Algorithm 1 requires the following assumption on f. This assumption is standard in Riemannian optimization [31, 32, 44]. The same assumption is used in subsequent studies [30, 33, 34]. Tang et al. also adopt this assumption in their framework [37].

Assumption 1. *The objective function f is continuously differentiable and bounded below on the manifold $\mathcal{M}$. Moreover, there exists a Lipschitz constant $L_1 > 0$ such that*

$$\left|\mathrm{D}(f \circ R_x)(t\eta)[\eta] - \mathrm{D}(f \circ R_x)(0_x)[\eta]\right| \leq L_1 t,$$

for all $x \in \mathcal{M}$, $\eta \in T_x\mathcal{M}$ with $\|\eta\|_x = 1$ and $t \geq 0$.

The following lemma derives a key property of the stepsize α_k that will be used to prove global convergence.

Lemma 1. *Suppose that Assumption 1 holds and the search direction η_k satisfies*

$$\langle g_k, \eta_k\rangle_{x_k} \leq -c_1\|g_k\|_{x_k}^2 \quad \text{and} \quad \|\eta_k\|_{x_k}^2 \leq (c_2 + c_3k)\|g_k\|_{x_k}^2, \tag{2.7}$$

where $c_1, c_2, c_3 > 0$. Then, the stepsize α_k generated by Algorithm 1 is bounded below, i.e.,

$$\alpha_k \geq \frac{(1-\sigma_2)c_1}{(c_2 + c_3k)L_1}.$$

Proof. By the curvature condition in the Riemannian Wolfe line search (2.3), we have

$$\langle g_{k+1}, \mathcal{T}_{\alpha_k \eta_k}(\eta_k)\rangle_{x_{k+1}} \geq \sigma_2 \langle g_k, \eta_k\rangle_{x_k}.$$

Combining this with Assumption 1, which gives

$$\langle g_{k+1}, \mathcal{T}_{\alpha_k \eta_k}(\eta_k)\rangle_{x_{k+1}} - \langle g_k, \eta_k\rangle_{x_k} \leq L_1 \alpha_k \|\eta_k\|_{x_k}^2,$$

we obtain

$$\alpha_k \geq \frac{(\sigma_2 - 1)\langle g_k, \eta_k\rangle_{x_k}}{L_1 \|\eta_k\|_{x_k}^2}.$$

Together with (2.7), this leads to

$$\alpha_k \geq \frac{(\sigma_2 - 1)\,\langle g_k, \eta_k\rangle_{x_k}}{L_1\,\|\eta_k\|_{x_k}^2} \geq \frac{(1-\sigma_2)\,c_1}{L_1} \cdot \frac{\|g_k\|_{x_k}^2}{\|\eta_k\|_{x_k}^2} \geq \frac{(1-\sigma_2)\,c_1}{(c_2 + c_3 k)\,L_1}.$$

This completes the proof. □

The first inequality in (2.7) indicates that η_k possesses the sufficient descent condition, which is universally valid in gradient-based algorithms. The second inequality is typically expressed as $\|\eta_k\|_{x_k}^2 \leq c_2 \|g_k\|_{x_k}^2$. In this paper, we introduce an appropriate modification that preserves the convergence guarantees while adapting it to a broader class of improved gradient-based algorithms. The specific construction of η_k satisfying the modified condition will be presented in Section 3.

The principal global convergence results of the Algorithm 1 are established in Theorem 1.

Theorem 1. *Suppose Assumption 1 holds and the search direction η_k satisfies* (2.7). *For the sequence $\{x_k\}$ produced by Algorithm 1, we have*

$$\liminf_{k\to\infty} \|g_k\|_{x_k} = 0.$$

Proof. From (2.7) and Lemma 1, it follows that

$$\begin{aligned}
\frac{\sigma_1\,(1-\sigma_2)\,c_1^2}{2\max(c_2, c_3)\,kL_1}\,\|g_k\|_{x_k}^2 &\leq \frac{\sigma_1\,(1-\sigma_2)\,c_1^2}{(c_2 + c_3 k)\,L_1}\,\|g_k\|_{x_k}^2 \\
&\leq \frac{-\sigma_1\,(1-\sigma_2)\,c_1\,\langle g_k, \eta_k\rangle_{x_k}}{(c_2 + c_3 k)\,L_1} \\
&\leq -\sigma_1 \alpha_k\,\langle g_k, \eta_k\rangle_{x_k}.
\end{aligned}$$

This result with (2.4) implies

$$\frac{\sigma_1\,(1-\sigma_2)\,c_1^2}{2\max(c_2, c_3)\,kL_1}\,\|g_k\|_{x_k}^2 \leq f(x_k) + \delta_k - f(x_{k+1}). \tag{2.8}$$

It follows that $k(f(x_k) + \delta_k - f(x_{k+1})) \geq 0$. From (2.4), (2.5), (2.7), and Lemma 1, we obtain

$$f(x_k) \leq C_{k-1} + \sigma_1 \alpha_{k-1} \langle g_{k-1}, \eta_{k-1}\rangle_{x_{k-1}} \leq C_{k-1},$$

which yields

$$C_k = \frac{t_{k-1}Q_{k-1}C_{k-1} + f(x_k)}{Q_k} \geq f(x_k).$$

For $k \geq 1$, (2.5) gives

$$0 \leq \delta_k \leq \frac{1}{k\log(k/d+12)},$$

which implies

$$0 \leq k\delta_k \leq \frac{1}{\log(k/d+12)} \to 0$$

as $k \to \infty$. Thus,

$$\lim_{k\to\infty} k\delta_k = 0.$$

Using this limit, we have

$$\begin{aligned}\liminf_{k\to\infty} k(f(x_k) - f(x_{k+1})) &\geq \liminf_{k\to\infty} k(f(x_k) + \delta_k - f(x_{k+1})) + \liminf_{k\to\infty}(-k\delta_k)\\ &= l \geq 0.\end{aligned}$$

Based on this, we proceed by contradiction to establish

$$\liminf_{k\to\infty} k\left(f\left(x_k\right) + \delta_k - f\left(x_{k+1}\right)\right) = l = 0.$$

Suppose $l > 0$ and set $\epsilon = \min\left\{1, \frac{l}{2}\right\} > 0$. Then, there exists $N \in \mathbb{N}^+$ such that for all $k > N$, $k\left(f\left(x_k\right) - f\left(x_{k+1}\right)\right) > \epsilon$, from which we deduce $f\left(x_{k+1}\right) < f\left(x_k\right) - \frac{\epsilon}{k}$. Since

$$\sum_{k=N+1}^{+\infty} \frac{1}{k} = +\infty,$$

we obtain

$$\lim_{k\to+\infty} f\left(x_k\right) = -\infty,$$

which contradicts the condition in Assumption 1 that f is bounded below. Thus,

$$\liminf_{k\to\infty} k\left(f\left(x_k\right) + \delta_k - f\left(x_{k+1}\right)\right) = l = 0.$$

From (2.8) and

$$\lim_{k\to+\infty} k\delta_k = 0,$$

we deduce

$$\liminf_{k\to\infty} \|g_k\|_{x_k} = 0.$$

This concludes the proof. □

3. Our method

This section describes the subspace minimization Riemannian CG method with nonmonotone Wolfe line search and a restart strategy. The search direction is first derived. The complete algorithmic framework is then presented, with detailed proof of global convergence.

3.1. The search direction η_k

In this subsection, building upon [25], we propose a new search direction that combines the dimension reduction technique in subspace minimization CG methods with the Riemannian BB stepsize.

The search direction η_k is determined by solving the following quadratic approximation problem constrained to the subspace spanned by g_k and s_{k-1}:

$$\min_{\eta \in \mathrm{Span}\{g_k, s_{k-1}\}} \langle g_k, \eta \rangle_{x_k} + \frac{1}{2} \langle \eta, B_k \eta \rangle_{x_k}, \tag{3.1}$$

where

$$s_{k-1} = \mathcal{T}_{\alpha_{k-1}\eta_{k-1}} (\alpha_{k-1}\eta_{k-1}) \in T_{x_k}\mathcal{M},$$

and B_k is a symmetric positive definite approximation to the Hessian of f at x_k that satisfies the secant condition $B_k s_{k-1} = y_{k-1}$, where

$$y_{k-1} = g_k - \mathcal{T}_{\alpha_{k-1}\eta_{k-1}} (g_{k-1}) \in T_{x_k}\mathcal{M}.$$

The subspace $\mathrm{Span}\{g_k, s_{k-1}\}$ is chosen because it preserves the CG update structure, exploits second-order information carried by s_{k-1}, and keeps the minimization computationally efficient (see [20, 25]). Set

$$\eta = \theta g_k + \beta s_{k-1}, \tag{3.2}$$

where $\theta, \beta \in \mathbb{R}$. We assume g_k and s_{k-1} are non-collinear throughout our discussion. Substituting (3.2) into (3.1) with $B_k s_{k-1} = y_{k-1}$ yields

$$\min_{\theta,\beta \in \mathbb{R}} \begin{pmatrix} \|g_k\|_{x_k}^2 \\ \langle g_k, s_{k-1} \rangle_{x_k} \end{pmatrix}^T \begin{pmatrix} \theta \\ \beta \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \theta & \beta \end{pmatrix} \begin{pmatrix} \rho_k & \langle g_k, y_{k-1} \rangle_{x_k} \\ \langle y_{k-1}, g_k \rangle_{x_k} & \langle s_{k-1}, y_{k-1} \rangle_{x_k} \end{pmatrix} \begin{pmatrix} \theta \\ \beta \end{pmatrix}, \tag{3.3}$$

where $\rho_k = \langle g_k, B_k g_k \rangle_{x_k}$. Define $\Delta_k = \rho_k \langle s_{k-1}, y_{k-1} \rangle_{x_k} - \langle g_k, y_{k-1} \rangle_{x_k}^2$. When $\Delta_k > 0$, problem (3.3) is strictly convex and admits a unique solution (θ_k, β_k) given explicitly by

$$\begin{aligned} \theta_k &= \frac{1}{\Delta_k} \left(\langle g_k, y_{k-1} \rangle_{x_k} \langle g_k, s_{k-1} \rangle_{x_k} - \langle s_{k-1}, y_{k-1} \rangle_{x_k} \|g_k\|_{x_k}^2 \right), \\ \beta_k &= \frac{1}{\Delta_k} \left(\langle g_k, y_{k-1} \rangle_{x_k} \|g_k\|_{x_k}^2 - \rho_k \langle g_k, s_{k-1} \rangle_{x_k} \right). \end{aligned}$$

Therefore,

$$\begin{aligned} \eta_k &= \theta_k g_k + \beta_k s_{k-1} \\ &= \frac{1}{\Delta_k} \Big[\left(\langle g_k, y_{k-1} \rangle_{x_k} \langle g_k, s_{k-1} \rangle_{x_k} - \langle s_{k-1}, y_{k-1} \rangle_{x_k} \|g_k\|_{x_k}^2 \right) g_k \\ &\quad + \left(\langle g_k, y_{k-1} \rangle_{x_k} \|g_k\|_{x_k}^2 - \rho_k \langle g_k, s_{k-1} \rangle_{x_k} \right) s_{k-1} \Big]. \end{aligned} \tag{3.4}$$

Building upon the foregoing analysis, we hereby present a novel search direction, which will be examined through four distinct cases:

Case 1: When $\langle g_k, s_{k-1}\rangle_{x_k}^2$ is sufficiently large, we naturally expect the current iterate x_k to be close to the exact point $\tilde{x}_k := R_{x_{k-1}}(\tilde{\alpha}_{k-1}\eta_{k-1})$, where $\tilde{\alpha}_{k-1} = \arg\min_{\alpha>0} f\left(R_{x_{k-1}}(\alpha\eta_{k-1})\right)$, since $\langle g_k, s_{k-1}\rangle_{x_k} = 0$ at $\tilde{x}_k$. Thus, if $\langle g_k, s_{k-1}\rangle_{x_k}$ is large, the next iteration should move backward along s_{k-1}. Conversely, if $-\langle g_k, s_{k-1}\rangle_{x_k}$ is large, the new step should proceed forward along s_{k-1}. The analysis demonstrates that $\theta = 0$ in (3.2). Solving (3.3) yields $\theta_k = 0$ and $\beta_k = -\frac{\langle g_k, s_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}$. Then,

$$\eta_k = -\frac{\langle g_k, s_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} s_{k-1}. \tag{3.5}$$

To guarantee that η_k retains the essential properties outlined in Section 3.3, we adopt (3.5) as the search direction under the conditions

$$\frac{\langle g_k, s_{k-1}\rangle_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k} \|g_k\|_{x_k}^2} \geq z_1 \quad \text{and} \quad \frac{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}{\|s_{k-1}\|_{x_k}^2} \geq \frac{z_2}{\sqrt{k}}, \tag{3.6}$$

where $0.8 < z_1 \leq 1$ and $0 < z_2 < 10^{-4}$.

Case 2: When the conditions

$$\frac{\langle g_k, s_{k-1}\rangle_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k} \|g_k\|_{x_k}^2} < z_1, \ \frac{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}{\|s_{k-1}\|_{x_k}^2} \geq \frac{z_2}{\sqrt{k}} \quad \text{and} \quad \frac{\|y_{k-1}\|_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} \leq z_3, \tag{3.7}$$

hold with $z_3 > 10^4$, the condition number of the Hessian matrix of f may not be large. In this case, we set B_k to be the identity operator scaled by the BB stepsize; specifically,

$$B_k = \frac{3}{2}\frac{\|y_{k-1}\|_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} \mathrm{id}_{T_{x_k}\mathcal{M}},$$

which corresponds to the Riemannian version of the formula proposed in [22]. The coefficient $\frac{3}{2}$ is adopted from [22], where it is shown to improve numerical performance by balancing the scaling between the gradient and curvature information. Then,

$$\rho_k = \frac{3}{2}\frac{\|y_{k-1}\|_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} \|g_k\|_{x_k}^2.$$

The search direction η_k is then given by

$$\begin{aligned}\eta_k = &\frac{\langle g_k, y_{k-1}\rangle_{x_k}\langle g_k, s_{k-1}\rangle_{x_k} - \langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2}{\frac{3}{2}\|y_{k-1}\|_{x_k}^2\|g_k\|_{x_k}^2 - \langle g_k, y_{k-1}\rangle_{x_k}^2} g_k \\ &+ \frac{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\langle g_k, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2 - \frac{3}{2}\|y_{k-1}\|_{x_k}^2\|g_k\|_{x_k}^2\langle g_k, y_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\left(\frac{3}{2}\|y_{k-1}\|_{x_k}^2\|g_k\|_{x_k}^2 - \langle g_k, s_{k-1}\rangle_{x_k}^2\right)} s_{k-1}.\end{aligned} \tag{3.8}$$

Case 3: If the conditions

$$\frac{\|y_{k-1}\|_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} > z_3 \quad \text{and} \quad \frac{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}{\|s_{k-1}\|_{x_k}^2} \geq \frac{z_2}{\sqrt{k}}$$

hold, the Hessian of f may exhibit a large condition number. Therefore, for the selected B_k in Case 2, it is no longer applicable. When

$$\frac{\left|\langle g_k, s_{k-1}\rangle_{x_k}\langle g_k, y_{k-1}\rangle_{x_k}\right|}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2} \leq z_4,$$

where $0 < z_4 \leq 10^{-4}$, we use

$$B_k = \mathrm{id}_{T_{x_k}\mathcal{M}} + \frac{y_{k-1}y_{k-1}^{\flat}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}.$$

Then,

$$\rho_k = \|g_k\|_{x_k}^2 + \frac{\langle g_k, y_{k-1}\rangle_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}.$$

Substituting ρ_k into (3.4) leads to

$$\begin{aligned}\eta_k =& \left(-1 + \frac{\langle g_k, y_{k-1}\rangle_{x_k}\langle g_k, s_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2}\right) g_k \\ &+ \left[\left(1 - \frac{\langle g_k, y_{k-1}\rangle_{x_k}\langle g_k, s_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2}\right)\frac{\langle g_k, y_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} - \frac{\langle g_k, s_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}\right] s_{k-1}.\end{aligned} \tag{3.9}$$

Therefore, if

$$\begin{aligned}&\frac{\langle g_k, s_{k-1}\rangle_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2} < z_1, \quad \frac{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}{\|s_{k-1}\|_{x_k}^2} \geq \frac{z_2}{\sqrt{k}}, \\ &\frac{\|y_{k-1}\|_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} > z_3, \quad \frac{\left|\langle g_k, s_{k-1}\rangle_{x_k}\langle g_k, y_{k-1}\rangle_{x_k}\right|}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2} \leq z_4,\end{aligned} \tag{3.10}$$

then the search direction is computed by (3.9).

Case 4: In all other cases, the direction is taken as $\eta_k = -g_k$. The same expression is generated by solving (3.3) under the conditions $\beta = 0$ and $B_k = \mathrm{id}_{T_{x_k}\mathcal{M}}$.

The parameters z_1, z_2, z_3, and z_4 serve as decision thresholds. Specifically, z_1 controls the relative magnitude of the gradient projection along the previous direction (Case 1); z_2 ensures sufficient positive curvature so that the denominators are well-defined (Cases 1–3); z_3 distinguishes well-conditioned from ill-conditioned curvature regimes (Cases 2 and 3); and z_4 limits the cross-term to avoid numerical instability (Case 3). Their specific values are determined empirically through preliminary numerical experiments and are fixed for all test problems in Section 5.

3.2. Algorithm description

Now, we present the framework of the subspace minimization Riemannian CG algorithm with nonmonotone Wolfe line search and a restart strategy. Prior to further analysis, key algorithmic details are elaborated. For convenience, let $\phi_k(\alpha) = f(R_{x_k}(\alpha\eta_k))$, where $\alpha > 0$. To measure the quality of the local quadratic approximation of $\phi_k(\alpha)$, we consider the ratio

$$r_k = \frac{\phi_k(0) - \phi_k(\alpha_k)}{q_k(0) - q_k(\alpha_k)}, \tag{3.11}$$

where

$$q_k(\alpha) := \left(\frac{\phi_k'(\alpha_k) - \phi_k'(0)}{2\alpha_k}\right)\alpha^2 + \phi_k'(0)\,\alpha + \phi_k(0)$$

is the quadratic function constructed by satisfying the interpolation conditions $q_k(0) = \phi_k(0)$, $q_k'(0) = \phi_k'(0)$, $q_k'(\alpha_k) = \phi_k'(\alpha_k)$. If r_k approaches 1, it indicates that the quadratic function q_k approximates the function ϕ_k, and otherwise, not. Similar to [47], if the condition $|r_k - 1| \leq z_5$ (where $z_5 > 0$) holds for multiple consecutive iterations, this indicates that the algorithm has likely entered a local quadratic region of the objective function. In this case, it is advisable to restart the algorithm along the negative gradient direction $-g_k$, and thus a restart should be performed. Algorithm 2 summarizes the proposed subspace minimization Riemannian CG method with nonmonotone Wolfe line search and a restart strategy.

Algorithm 2 A subspace minimization Riemannian CG method with nonmonotone Wolfe line search and a restart strategy (SRCG-NWR).

Step 0: Choose an initial iterate $x_0 \in \mathcal{M}$; parameters $\sigma_1 \in (0, 1)$, $\sigma_2 \in (\sigma_1, 1)$, z_1, z_2, z_3, z_4, z_5; positive integers M_R, M_Q; a tolerance $\varepsilon \geq 0$. Set $i_R := 0$, $i_Q := 0$, $n_{cg} := 0$, $\eta_0 = -g_0$ and $k := 0$.

Step 1: If $\|g_k\|_{x_k} \leq \varepsilon$, then **stop**.

Step 2: Select $\alpha_k^{It} > 0$. Determine α_k satisfying the nonmonotone Wolfe line search (2.3) and (2.4) with α_k^{It}.

Step 3: Compute $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$ and g_{k+1}. Set $i_R = i_R + 1$. Compute r_k by (3.11). If $|r_k - 1| \leq z_5$ or $\left|f(x_{k+1}) - f(x_k) - \frac{\alpha_k}{2}\left(\phi_k'(\alpha_k) + \phi_k'(0)\right)\right| \leq z_5$, then set $i_Q = i_Q + 1$; otherwise, set $i_Q := 0$.

Step 4: If $n_{cg} = M_R$ or ($i_Q = M_Q$ and $i_R \neq i_Q$), then set $n_{cg} = 0$, $i_R = 0$, $i_Q = 0$ and $\eta_{k+1} = -g_{k+1}$, and go to Step 6.

Step 5: Compute the search direction.

Case 1: If condition (3.6) holds, then set $n_{cg} = n_{cg} + 1$, and compute the search direction η_{k+1} by (3.5).

Case 2: If condition (3.7) holds, then set $n_{cg} = n_{cg} + 1$, and compute the search direction η_{k+1} by (3.8).

Case 3: If condition (3.10) holds, then set $n_{cg} = n_{cg} + 1$, and compute the search direction η_{k+1} by (3.9).

Case 4: Otherwise, set $n_{cg} = 0$, $i_R = 0$, $i_Q = 0$ and $\eta_{k+1} = -g_{k+1}$.

Step 6: Compute δ_{k+1} by (2.5) and set $k := k + 1$, go to Step 1.

Remark 1. *Some comments on Algorithm 2 are given below:*

(a) In Step 0, i_R records the iteration count after restart, i_Q is the indicator for the quadratic approximation accuracy of ϕ, M_R is the maximum restart iteration threshold, and M_Q is the minimum

restart threshold to guarantee consecutive quadratic approximation accuracy of ϕ.

(b) The initial stepsize α_k^{It} may be selected either as a fixed value or by extending the scheme in [25] to the Riemannian setting. Since the generalization is straightforward, we omit further discussion here.

(c) According to (3.11), we have

$$|r_k - 1| = \left|\frac{\phi_k(0) - \phi_k(\alpha_k)}{q_k(0) - q_k(\alpha_k)} - 1\right| = \left|\frac{q_k(\alpha_k) - \phi_k(\alpha_k)}{q_k(0) - q_k(\alpha_k)}\right| = \left|\frac{f(x_{k+1}) - f(x_k) - \frac{\alpha_k}{2}\left(\phi_k'(\alpha_k) + \phi_k'(0)\right)}{q_k(0) - q_k(\alpha_k)}\right|.$$

From this expression, we observe that $|r_k - 1|$ may become large when the denominator $|q_k(0) - q_k(\alpha_k)|$ is very small. In our experiments, we only require that $\phi_k(\alpha_k)$ be sufficiently close to $q_k(\alpha_k)$ to be acceptable. Therefore, we introduce the condition

$$|q_k(\alpha_k) - \phi_k(\alpha_k)| = \left|f(x_{k+1}) - f(x_k) - \frac{\alpha_k}{2}\left(\phi_k'(\alpha_k) + \phi_k'(0)\right)\right| \leq z_5$$

in Step 3 of Algorithm 2.

3.3. Convergence analysis

The global convergence of Algorithm 2 is proved in this section. For this purpose, a fundamental assumption proposed in [49] is first introduced.

Assumption 2. *For a vector transport $\mathcal{T}$ corresponding to retraction R, there exists a positive constant L_2 such that for any $x \in \mathcal{M}$ and $\eta \in T_x\mathcal{M}$,*

$$\left\|g_{R_x(\eta)} - \mathcal{T}_\eta(g_x)\right\|_{R_x(\eta)} \leq L_2 \left\|\mathcal{T}_\eta(\eta)\right\|_{R_x(\eta)}.$$

Assumption 2 is satisfied for many optimization problems on commonly used manifolds, such as the sphere, the Stiefel manifold, and the Grassmann manifold, provided that the objective function has a Lipschitz continuous Riemannian gradient and standard smooth retractions and vector transports are employed [29, 46]. Under Assumption 2, for all $k \geq 1$,

$$\|y_{k-1}\|_{x_k} \leq L_2 \|s_{k-1}\|_{x_k}. \tag{3.12}$$

In the following, two lemmas are presented to demonstrate that the search direction η_k satisfies Eq (2.7).

Lemma 2. *If Assumption 2 is satisfied, then the search direction η_k generated by Algorithm 2 satisfies*

$$\langle g_k, \eta_k \rangle_{x_k} \leq -c_1 \|g_k\|_{x_k}^2, \tag{3.13}$$

where $c_1 = \frac{2}{3z_3}$.

Proof. The search direction strategy (Section 3.1) admits four cases:

Case 1: If η_k is calculated using (3.5), then $\langle g_k, \eta_k \rangle_{x_k} = -\frac{\langle g_k, s_{k-1} \rangle_{x_k}^2}{\langle s_{k-1}, y_{k-1} \rangle_{x_k}} \leq -z_1 \|g_k\|^2$.

Case 2: Combining (3.4) with the definition of Δ_k yields

$$\begin{aligned}\langle g_k, \eta_k\rangle_{x_k} &\leq -\frac{\|g_k\|_{x_k}^4}{\Delta_k}\left[\langle s_{k-1}, y_{k-1}\rangle_{x_k} - \frac{2\langle g_k, y_{k-1}\rangle_{x_k}\langle g_k, s_{k-1}\rangle_{x_k}}{\|g_k\|_{x_k}^2} + \rho_k\left(\frac{\langle g_k, s_{k-1}\rangle_{x_k}}{\|g_k\|_{x_k}^2}\right)^2\right]\\ &= -\frac{\|g_k\|_{x_k}^4}{\Delta_k}\left[\langle s_{k-1}, y_{k-1}\rangle_{x_k} - \frac{\langle g_k, y_{k-1}\rangle_{x_k}^2}{\rho_k} + \left(\frac{\sqrt{\rho_k}\langle g_k, s_{k-1}\rangle_{x_k}}{\|g_k\|_{x_k}^2} - \frac{\langle g_k, y_{k-1}\rangle_{x_k}}{\sqrt{\rho_k}}\right)^2\right]\\ &\leq -\frac{\|g_k\|_{x_k}^4}{\Delta_k}\left(\langle s_{k-1}, y_{k-1}\rangle_{x_k} - \frac{\langle g_k, y_{k-1}\rangle_{x_k}^2}{\rho_k}\right)\\ &= -\frac{\|g_k\|_{x_k}^4}{\Delta_k}\cdot\frac{\Delta_k}{\rho_k} = -\frac{\|g_k\|_{x_k}^4}{\rho_k}.\end{aligned}$$

Therefore, for η_k computed by (3.8), we obtain

$$\langle g_k, \eta_k\rangle_{x_k} \leq -\frac{\|g_k\|_{x_k}^4}{\rho_k} = -\frac{2}{3}\frac{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}{\|y_{k-1}\|_{x_k}^2}\|g_k\|_{x_k}^2 \leq -\frac{2}{3z_3}\|g_k\|_{x_k}^2.$$

Case 3: Computing η_k through (3.9) yields

$$\langle g_k, \eta_k\rangle_{x_k} = -\|g_k\|_{x_k}^2\left[1 - 2\frac{\langle g_k, y_{k-1}\rangle_{x_k}\langle g_k, s_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2} + \frac{\langle g_k, s_{k-1}\rangle_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2} + \frac{\langle g_k, y_{k-1}\rangle_{x_k}^2\langle g_k, s_{k-1}\rangle_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}^2\|g_k\|_{x_k}^4}\right].$$

Since the last two terms inside the brackets are nonnegative, we have

$$\langle g_k, \eta_k\rangle_{x_k} \leq -\|g_k\|_{x_k}^2\left[1 - 2\frac{\langle g_k, y_{k-1}\rangle_{x_k}\langle g_k, s_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2}\right].$$

By the condition in (3.10), we have

$$\left|\frac{\langle g_k, y_{k-1}\rangle_{x_k}\langle g_k, s_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}\|g_k\|_{x_k}^2}\right| \leq z_4,$$

which yields

$$\langle g_k, \eta_k\rangle_{x_k} \leq -(1 - 2z_4)\|g_k\|_{x_k}^2.$$

Case 4: The result $\langle g_k, \eta_k\rangle_{x_k} = -\|g_k\|_{x_k}^2$ is directly derivable when $\eta_k = -g_k$.

To conclude, defining $c_1 = \min\left\{\frac{2}{3z_3}, 1 - 2z_4, z_1\right\}$, and considering $0 < z_4 \leq 10^{-4}$, $0.8 < z_1 \leq 1$ and $z_3 > 10^4$, it follows that $c_1 = \frac{2}{3z_3}$. Consequently, Eq (3.13) holds, which completes the proof. □

Lemma 3. *Suppose that Assumption 2 holds, then the search direction η_k generated by Algorithm 2 satisfies*

$$\|\eta_k\|_{x_k}^2 \leq (c_2 + c_3 k)\|g_k\|_{x_k}^2, \tag{3.14}$$

where $c_2 = (1 + z_4)^2$ and $c_3 = \max\left\{\frac{81}{z_2^2}, \frac{(1+L_2+z_4L_2)(3+L_2+2z_4+z_4L_2)}{z_2^2}\right\}$.

Proof. Four cases are considered in the following discussion.

Case 1: In the case where η_k is computed using (3.5), we obtain

$$\|\eta_k\|_{x_k}^2 = \frac{\langle g_k, s_{k-1}\rangle_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}^2} \|s_{k-1}\|_{x_k}^2 \leq \frac{\|s_{k-1}\|_{x_k}^4}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}^2} \|g_k\|_{x_k}^2 \leq \frac{k}{z_2^2} \|g_k\|_{x_k}^2.$$

Case 2: Using (3.8) and

$$\frac{3}{2} \|y_{k-1}\|_{x_k}^2 \|g_{k-1}\|_{x_k}^2 - \langle g_k, y_{k-1}\rangle_{x_k}^2 \geq \frac{1}{2} \|y_{k-1}\|_{x_k}^2 \|g_k\|_{x_k}^2$$

results in

$$\begin{aligned}
\|\eta_k\|_{x_k} \leq & \frac{2\left(\langle g_k, y_{k-1}\rangle_{x_k} \langle g_k, s_{k-1}\rangle_{x_k} - \langle s_{k-1}, y_{k-1}\rangle_{x_k} \|g_k\|_{x_k}^2\right)}{\|y_{k-1}\|_{x_k}^2 \|g_k\|_{x_k}} \\
& + \frac{\left(2\langle s_{k-1}, y_{k-1}\rangle_{x_k} \langle g_k, y_{k-1}\rangle_{x_k} \|g_k\|_{x_k}^2 + 3\|y_{k-1}\|_{x_k}^2 \|g_k\|_{x_k}^2 \langle g_k, s_{k-1}\rangle_{x_k}\right) \|s_{k-1}\|_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k} \|y_{k-1}\|_{x_k}^2 \|g_k\|_{x_k}^2} \\
\leq & \frac{2\langle g_k, y_{k-1}\rangle_{x_k} \langle g_k, s_{k-1}\rangle_{x_k}}{\|y_{k-1}\|_{x_k}^2 \|g_k\|_{x_k}} + \frac{2\|g_k\|_{x_k} \|s_{k-1}\|_{x_k}}{\|y_{k-1}\|_{x_k}} + \frac{2\|g_k\|_{x_k} \|s_{k-1}\|_{x_k}}{\|y_{k-1}\|_{x_k}} + \frac{3\|s_{k-1}\|_{x_k}^2 \|g_k\|_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} \\
\leq & \frac{4\|s_{k-1}\|_{x_k} \|g_k\|_{x_k}}{\|y_{k-1}\|_{x_k}} + \frac{2\|s_{k-1}\|_{x_k} \|g_k\|_{x_k}}{\|y_{k-1}\|_{x_k}} + \frac{3\|s_{k-1}\|_{x_k}^2 \|g_k\|_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} \\
= & \left(6\frac{\|s_{k-1}\|_{x_k}}{\|y_{k-1}\|_{x_k}} + 3\frac{\|s_{k-1}\|_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}\right) \|g_k\|_{x_k} \\
\leq & \left(\frac{6\sqrt{k}}{z_2} + \frac{3\sqrt{k}}{z_2}\right) \|g_k\|_{x_k} = \frac{9\sqrt{k}}{z_2} \|g_k\|_{x_k}.
\end{aligned}$$

Hence, $\|\eta_k\|_{x_k}^2 \leq \frac{81k}{z_2^2} \|g_k\|_{x_k}^2$.

Case 3: From (3.9) and (3.12), it follows that

$$\begin{aligned}
\|\eta_k\|_{x_k} \leq & \|g_k\|_{x_k} + \frac{\left|\langle g_k, y_{k-1}\rangle_{x_k} \langle g_k, s_{k-1}\rangle_{x_k}\right|}{\langle s_{k-1}, y_{k-1}\rangle_{x_k} \|g_k\|_{x_k}^2} \|g_k\|_{x_k} \\
& + \left[\left(1 + \frac{\left|\langle g_k, y_{k-1}\rangle_{x_k} \langle g_k, s_{k-1}\rangle_{x_k}\right|}{\langle s_{k-1}, y_{k-1}\rangle_{x_k} \|g_k\|_{x_k}^2}\right) \frac{\langle g_k, y_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} + \frac{\langle g_k, s_{k-1}\rangle_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}\right] \|s_{k-1}\|_{x_k} \\
\leq & (1 + z_4) \|g_k\|_{x_k} + \left[(1 + z_4) \frac{\|g_k\|_{x_k} \|y_{k-1}\|_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} + \frac{\|g_k\|_{x_k} \|s_{k-1}\|_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}\right] \|s_{k-1}\|_{x_k} \\
\leq & (1 + z_4) \|g_k\|_{x_k} + \left[(1 + z_4) \frac{L_2 \|g_k\|_{x_k} \|s_{k-1}\|_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} + \frac{\|g_k\|_{x_k} \|s_{k-1}\|_{x_k}}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}\right] \|s_{k-1}\|_{x_k} \\
= & (1 + z_4) \|g_k\|_{x_k} + \left[(1 + L_2 + z_4 L_2) \frac{\|s_{k-1}\|_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}}\right] \|g_k\|_{x_k}
\end{aligned}$$

$$\leq \left[(1+z_4) + (1+L_2+z_4L_2)\,\frac{\sqrt{k}}{z_2}\right]\|g_k\|_{x_k}.$$

Therefore,

$$\begin{aligned}\|\eta_k\|_{x_k}^2 &\leq \left[(1+z_4)^2 + \frac{((1+z_4)\,L_2+1)^2}{z_2^2}k + 2\,(1+z_4)\,(1+L_2+z_4L_2)\,\frac{\sqrt{k}}{z_2}\right]\|g_k\|_{x_k}^2\\ &\leq \left[(1+z_4)^2 + \frac{(1+L_2+z_4L_2)\,(3+L_2+2z_4+z_4L_2)}{z_2^2}k\right]\|g_k\|_{x_k}^2,\end{aligned}$$

where the last inequality follows from

$$\frac{((1+z_4)\,L_2+1)^2}{z_2^2}k + 2(1+z_4)(1+L_2+z_4L_2)\frac{\sqrt{k}}{z_2} \leq \frac{(1+L_2+z_4L_2)(3+L_2+2z_4+z_4L_2)}{z_2^2}k,$$

which holds for $k \geq 1$ and $0 < z_2 < 10^{-4}$ (since $\sqrt{k} \leq k$).

Case 4: For $\eta_k = -g_k$, we have $\|\eta_k\|_{x_k}^2 = \|g_k\|_{x_k}^2$.

Let

$$c_2 = \max\left\{1, (1+z_4)^2\right\} \text{ and } c_3 = \max\left\{\frac{1}{z_2^2}, \frac{81}{z_2^2}, \frac{(1+L_2+z_4L_2)\,(3+L_2+2z_4+z_4L_2)}{z_2^2}\right\}.$$

The conditions $0 < z_4 \leq 10^{-4}$ and $0 < z_2 \leq 10^{-4}$ lead to

$$c_2 = (1+z_4)^2 \text{ and } c_3 = \max\left\{\frac{81}{z_2^2}, \frac{(1+L_2+z_4L_2)\,(3+L_2+2z_4+z_4L_2)}{z_2^2}\right\},$$

thereby verifying (3.14). □

Lemmas 2 and 3 imply that the search direction η_k generated by Algorithm 2 satisfies (2.7). Thus, the following conclusion can be readily derived from Theorem 1.

Theorem 2. *Suppose that Assumptions 1 and 2 hold. Then, the sequence $\{x_k\}$ generated by Algorithm 2 satisfies*

$$\liminf_{k\to\infty} \|g_k\|_{x_k} = 0.$$

4. R-linear convergence

This section develops the R-linear convergence analysis, which relies on the uniformly retraction-convexity (Definition 1) and several assumptions [50, 51]. In contrast to the convergence analysis (Theorem 2), this additional convexity-type condition is needed for establishing the local convergence rate.

In the convergence analysis, the infinite sequences $\{x_k\}$, $\{\alpha_k\}$, and $\{\eta_k\}$ are generated by Algorithm 2. The point x^* represents a local minimizer of f on a bounded set $\mathcal{N}_0 \subset \mathcal{M}$.

Definition 1. *[52] A function $f\colon \mathcal{M} \to \mathbb{R}$ is called uniformly retraction-convex on a subset $S \subset \mathcal{M}$ with respect to retraction R if the function $m_{x,\eta}(t) := f(R_x(t\eta))$ is uniformly convex. Specifically, there exists a constant $c_4 > 0$ such that*

$$\alpha m_{x,\eta}(t_1) + (1-\alpha) m_{x,\eta}(t_2) - m_{x,\eta}(t_2 + \alpha(t_1 - t_2)) \geq \frac{1}{2c_4}\alpha(1-\alpha)(t_1 - t_2)^2,$$

for all $x \in S$, all $\eta \in T_x\mathcal{M}$, and $\|\eta\|_x = 1$, $\alpha \in (0,1)$ and $t_1, t_2 \geq 0$ such that $R_x(\tau\eta) \in S$ for all $\tau \in [0, \max(t_1, t_2)]$.

Assumption 3. *The objective function f is uniformly retraction-convex with respect to the retraction R in $\mathcal{N}_0$.*

The definition of uniformly retraction-convex extends the corresponding concept from standard Euclidean space to Riemannian manifolds. It is easy to see that when the function is defined on Euclidean space, uniform retraction-convexity reduces to uniform convexity. For a detailed analysis of functions satisfying Assumption 3, see [50]. Moreover, for the neighborhood of x^*, the following conditions are satisfied.

Let U be a neighborhood of x^* and $\rho > 0$ such that for any $y \in U$, $U \subset R_y\left(\mathbb{B}\left(0_y, \rho\right)\right)$ and $R_y(\cdot)$ is a diffeomorphism on $\mathbb{B}\left(0_y, \rho\right)$, where

$$\mathbb{B}\left(0_y, \rho\right) = \left\{\eta_y \in T_y\mathcal{M} : \left\|\eta_y\right\|_y < \rho\right\}.$$

The existence of such ρ-totally retractive neighborhoods of x^* is established in [53, Section 3.3]. Further, assume that U is an R-star shaped neighborhood of x^*, meaning that $R_{x^*}\left(tR_{x^*}^{-1}(x)\right) \in U$ for all $x \in U$ and $t \in [0,1]$.

Assumption 4. *The iterates x_k stay continuously in a neighborhood U of x^*, which ensures that $R_{x_k}(t\eta_k) \in U \subset \mathcal{N}_0$ for all $t \in [0, \alpha_k]$.*

Assumption 5. *The vector transport $\mathcal{T}$ is isometric, i.e., for all $\eta, \xi, \zeta \in T_x\mathcal{M}$,*

$$\left\langle \mathcal{T}_\eta(\xi), \mathcal{T}_\eta(\zeta) \right\rangle_{R_x(\eta)} = \langle \xi, \zeta \rangle_x. \tag{4.1}$$

Assumption 4 is essential. For instance, on the unit sphere with exponential retraction, it is possible that $x_k = x_{k+1}$ while $\|\alpha_k\eta_k\|_{x_k} = 2\pi$. For details, see [50, 53]. For Assumption 5, there exist retractions satisfying the isometric condition (4.1), such as on the Stiefel and Grassmann manifolds [54]. The following analysis depends on Assumptions 3-5, which are standard in the local convergence analysis of Riemannian optimization algorithms. Their specific contents and application backgrounds are fully discussed and validated in [50, 52, 53]. Next, we introduce a key lemma for establishing R-linear convergence.

Lemma 4. *If Assumption 3 holds, then there exists a constant $c_4 > 0$ such that*

$$\left\langle g_{R_x(t\eta)}, \mathcal{T}_{t\eta}(t\eta) \right\rangle_{R_x(t\eta)} - \langle g_x, t\eta \rangle_x \geq \frac{1}{c_4} t^2$$

for all $x \in \mathcal{M}$, $\eta \in T_x\mathcal{M}$ with $\|\eta\|_x = 1$ and $t \geq 0$.

Proof. Consider the function $m_{x,\eta}(t) = f(R_x(t\eta))$ for $x \in \mathcal{M}$, $\eta \in T_x\mathcal{M}$ with $\|\eta\|_x = 1$, and $t \geq 0$. By the uniform retraction-convexity of f (Definition 1), there exists a constant $c_4 > 0$ such that for all $\alpha \in (0, 1)$,

$$\alpha m_{x,\eta}(0) + (1-\alpha) m_{x,\eta}(t) - m_{x,\eta}(t + \alpha(0-t)) \geq \frac{1}{2c_4}\alpha(1-\alpha)t^2.$$

Rearranging the terms gives

$$m_{x,\eta}(0) - m_{x,\eta}(t) \geq \frac{m_{x,\eta}(t + \alpha(0-t)) - m_{x,\eta}(t)}{\alpha} + \frac{1-\alpha}{2c_4}t^2.$$

Taking the limit $\alpha \to 0^+$ on both sides of the above inequality, and noting that

$$\lim_{\alpha \to 0^+} \frac{m_{x,\eta}(t + \alpha(0-t)) - m_{x,\eta}(t)}{\alpha} = -t\langle g_{R_x(t\eta)}, \mathrm{D}R_x(t\eta)[\eta]\rangle_{R_x(t\eta)},$$

we obtain

$$m_{x,\eta}(0) - m_{x,\eta}(t) \geq -t\langle g_{R_x(t\eta)}, \mathcal{T}_{t\eta}(\eta)\rangle_{R_x(t\eta)} + \frac{1}{2c_4}t^2.$$

Similarly, applying the same argument with 0 and t interchanged yields

$$m_{x,\eta}(t) - m_{x,\eta}(0) \geq t\langle g_x, \eta\rangle_x + \frac{1}{2c_4}t^2.$$

Adding the two inequalities gives

$$\left\langle g_{R_x(t\eta)}, \mathcal{T}_{t\eta}(t\eta)\right\rangle_{R_x(t\eta)} - \langle g_x, t\eta\rangle_x \geq \frac{1}{c_4}t^2.$$

which completes the proof. □

We now present the main result, which establishes the R-linear convergence rate of Algorithm 2.

Theorem 3. *Under Assumptions 1–5, given that f has a unique minimizer x^*, $t_{\max} < 1$, and $\alpha_k \leq \mu_{\max}$ for some $\mu_{\max} > 0$ and all k, there exists $\theta \in (0, 1)$ such that*

$$f(x_k) - f(x^*) \leq \theta^k (f(x_0) - f(x^*)). \tag{4.2}$$

Proof. By Lemma 4 and the isometric property of $\mathcal{T}$, it follows that

$$\begin{aligned}
\langle y_{k-1}, s_{k-1}\rangle_{x_k} &= \left\langle g_k - \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(g_{k-1}), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\alpha_{k-1}\eta_{k-1})\right\rangle_{x_k} \\
&= \left\langle g_k, \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}(\alpha_{k-1}\eta_{k-1})\right\rangle_{x_k} - \langle g_{k-1}, \alpha_{k-1}\eta_{k-1}\rangle_{x_{k-1}} \\
&= \left\langle g_k, \mathcal{T}_{\alpha_{k-1}\|\eta_{k-1}\|_{x_{k-1}} \frac{\eta_{k-1}}{\|\eta_{k-1}\|_{x_{k-1}}}}\left(\alpha_{k-1}\|\eta_{k-1}\|_{x_{k-1}} \frac{\eta_{k-1}}{\|\eta_{k-1}\|_{x_{k-1}}}\right)\right\rangle_{x_k} \\
&\quad - \left\langle g_{k-1}, \alpha_{k-1}\|\eta_{k-1}\|_{x_{k-1}} \frac{\eta_{k-1}}{\|\eta_{k-1}\|_{x_{k-1}}}\right\rangle_{x_{k-1}} \\
&\geq \frac{1}{c_4}\alpha_{k-1}^2 \|\eta_{k-1}\|_{x_{k-1}}^2 = \frac{1}{c_4}\|s_{k-1}\|_{x_k}^2.
\end{aligned}$$

This implies that

$$\frac{\|s_{k-1}\|_{x_k}^2}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}} \leq c_4, \quad \frac{\|s_{k-1}\|_{x_k}^4}{\langle s_{k-1}, y_{k-1}\rangle_{x_k}^2} \leq c_4^2 \text{ and } \frac{\|s_{k-1}\|_{x_k}}{\|y_{k-1}\|_{x_k}} \leq c_4.$$

From this and the proof of Lemma 3, we conclude that there exists $c > 0$ satisfying $\|\eta_k\|_{x_k} \leq c\,\|g_k\|_{x_k}$. Note that the constant c here is independent of k. Together with Lemma 1 and Eq (3.13), this yields

$$\alpha_k \geq \frac{(\sigma_2 - 1)\,\langle g_k, \eta_k\rangle_{x_k}}{L_1\,\|\eta_k\|_{x_k}^2} \geq \frac{(1-\sigma_2)\,c_1\,\|g_k\|_{x_k}^2}{L_1\,\|\eta_k\|_{x_k}^2} \geq \frac{(1-\sigma_2)\,c_1}{L_1 c^2} \triangleq \bar{\alpha}. \tag{4.3}$$

Furthermore, the application of (2.4) and (2.5) leads to

$$f(x_{k+1}) \leq f(x_k) + \delta_k + \sigma_1\alpha_k\,\langle g_k, \eta_k\rangle_{x_k} \leq C_k + \sigma_1\alpha_k\,\langle g_k, \eta_k\rangle_{x_k},$$

while the combination with (3.13) and (4.3) establishes

$$f(x_{k+1}) \leq C_k - q\,\|g_k\|_{x_k}^2, \tag{4.4}$$

where $q = \sigma_1\bar{\alpha}c_1$. Based on Assumption 2, along with the condition $\|\eta_k\|_{x_k} \leq c\,\|g_k\|_{x_k}$ and the isometric property of $\mathcal{T}$, we derive

$$\begin{aligned}
\|g_{k+1}\|_{x_{k+1}} &\leq \left\|g_{k+1} - \mathcal{T}_{\alpha_k\eta_k}(g_k)\right\|_{x_{k+1}} + \left\|\mathcal{T}_{\alpha_k\eta_k}(g_k)\right\|_{x_{k+1}} \\
&\leq L_2\alpha_k\left\|\mathcal{T}_{\alpha_k\eta_k}(\eta_k)\right\|_{x_{k+1}} + \|g_k\|_{x_k} \\
&= L_2\alpha_k\,\|\eta_k\|_{x_k} + \|g_k\|_{x_k} \\
&\leq (1 + \mu_{\max}L_2 c)\,\|g_k\|_{x_k} = \bar{b}\,\|g_k\|_{x_k},
\end{aligned} \tag{4.5}$$

where $\bar{b} = 1 + \mu_{\max}L_2 c$.

Now we prove the conclusion

$$C_{k+1} - f(x^*) \leq \theta\,(C_k - f(x^*)), \tag{4.6}$$

where $\theta = 1 - qb\,(1 - t_{\max})$ and $b = \frac{1}{q + c_4\bar{b}^2}$. Two cases are considered:

(i) $\|g_k\|_{x_k}^2 \geq b\,(C_k - f(x^*))$. By (2.6) and (4.4), it follows that

$$\begin{aligned}
C_{k+1} - f(x^*) &= \frac{t_k Q_k C_k + f(x_{k+1})}{Q_{k+1}} - f(x^*) \\
&= \frac{t_k Q_k\,(C_k - f(x^*)) + (f(x_{k+1}) - f(x^*))}{Q_{k+1}} \\
&\leq \frac{t_k Q_k\,(C_k - f(x^*)) + (C_k - f(x^*)) - q\,\|g_k\|_{x_k}^2}{Q_{k+1}} \\
&= C_k - f(x^*) - \frac{q\,\|g_k\|_{x_k}^2}{Q_{k+1}}.
\end{aligned}$$

It follows from [52] that

$$Q_{k+1} \leq \frac{1}{1 - t_{\max}}.$$

Thus,

$$C_{k+1} - f(x^*) \leq C_k - f(x^*) - q(1 - t_{\max}) \|g_k\|_{x_k}^2.$$

Given that

$$\|g_k\|_{x_k}^2 \geq b(C_k - f(x^*)),$$

inequality (4.6) is satisfied.

(ii) $\|g_k\|_{x_k}^2 < b(C_k - f(x^*))$. Define

$$p_{k+1} = \left\|R_{x^*}^{-1}(x_{k+1})\right\|_{x^*} \text{ and } \zeta_{k+1} = \frac{R_{x^*}^{-1}(x_{k+1})}{p_{k+1}}.$$

Substituting $x = x^*$, $t = p_{k+1}$, and $\eta = \zeta_{k+1}$ into Lemma 4, and noting that x^* is a minimizer, we obtain

$$\left\langle g_{k+1}, \mathcal{T}_{R_{x^*}^{-1}(x_{k+1})}\left(R_{x^*}^{-1}(x_{k+1})\right)\right\rangle_{x_{k+1}} \geq \frac{1}{c_4} p_{k+1}^2.$$

Therefore, $p_{k+1} \leq c_4 \|g_{k+1}\|_{x_{k+1}}$. Define $m_{x^*,\zeta_{k+1}}(t) = f(R_{x^*}(t\zeta_{k+1}))$, $t \in [0, p_{k+1}]$. Since f is uniformly retraction-convex, $m_{x^*,\zeta_{k+1}}(t)$ is a uniformly convex function of t. Consequently, the derivative $m'_{x^*,\zeta_{k+1}}(t)$ is monotonically increasing for $t \in [0, p_{k+1}]$ with $m'_{x^*,\zeta_{k+1}}(0) = 0$. Therefore, $m'_{x^*,\zeta_{k+1}}(t)$ attains its maximum at $t = p_{k+1}$. From the above analysis, it follows that

$$\begin{aligned} f(x_{k+1}) - f(x^*) &= m_{x^*,\zeta_{k+1}}(p_{k+1}) - m_{x^*,\zeta_{k+1}}(0) \\ &= \int_0^{p_{k+1}} m'_{x^*,\zeta_{k+1}}(t)\, dt \\ &\leq m'_{x^*,\zeta_{k+1}}(p_{k+1})\, p_{k+1} \\ &= \left\langle g_{k+1}, \mathcal{T}_{R_{x^*}^{-1}(x_{k+1})}(\zeta_k)\right\rangle_{x_{k+1}} p_{k+1} \\ &\leq c_4 \|g_{k+1}\|_{x_{k+1}}^2. \end{aligned}$$

These results, combined with (4.5) and the condition $\|g_k\|_{x_k}^2 < b(C_k - f(x^*))$, imply

$$f(x_{k+1}) - f(x^*) \leq c_4 \|g_{k+1}\|_{x_{k+1}}^2 \leq c_4 b\bar{b}^2 (C_k - f(x^*)).$$

Thus,

$$\begin{aligned} C_{k+1} - f(x^*) &= \frac{t_k Q_k (C_k - f(x^*)) + (f(x_{k+1}) - f(x^*))}{Q_{k+1}} \\ &\leq \frac{t_k Q_k (C_k - f(x^*)) + c_4 b\bar{b}^2 (C_k - f(x^*))}{Q_{k+1}} \\ &= \left(1 - \frac{1 - c_4 b\bar{b}^2}{Q_{k+1}}\right)(C_k - f(x^*)) \\ &\leq \left[1 - (1 - t_{\max})\left(1 - c_4 b\bar{b}^2\right)\right](C_k - f(x^*)). \end{aligned}$$

Since $1 - c_4 b\bar{b}^2 = qb$, inequality (4.6) is valid.

The recursive relation

$$C_{k+1} - f(x^*) \leq \theta(C_k - f(x^*)) \leq \cdots \leq \theta^{k+1}(C_0 - f(x^*))$$

follows from (4.6). From Eq (4.4), we obtain

$$C_{k+1} = \frac{t_k Q_k C_k + f_{k+1}}{Q_{k+1}} \geq f(x_{k+1}).$$

Therefore, inequality (4.2) holds. The proof is complete. □

5. Numerical experiments

This section evaluates the numerical performance of the proposed SRCG-NWR method. Section 5.1 describes the test problems and parameter settings. Section 5.2 compares SRCG-NWR with eleven existing Riemannian optimization algorithms.

5.1. Test problems and experimental setup

Seven problems are selected for numerical testing, the details of which are summarized in Table 1.

Table 1. Test problems.

Prob.	Size	Problem description and source	$\mathcal{M}$
P1	$n = 100$ $n = 500$ $n = 1000$	The Rayleigh quotient problem [29]: given an $n \times n$ symmetric matrix, find its smallest eigenvalue.	Sphere
P2	$n = 50, p = 5$ $n = 100, p = 10$ $n = 200, p = 20$	Optimizing the Rayleigh quotient on the set of p-dimensional subspaces of $\mathbb{R}^n$ [29].	Grassmann
P3	$n = 100, p = 5$ $n = 200, p = 10$ $n = 500, p = 20$	Given a matrix, minimize the Brockett cost function over the set of all $n \times p$ orthonormal matrices to extract its ordered eigenvectors [29].	Stiefel
P4	$n = 12, p = 4, N = 16$ $n = 12, p = 4, N = 64$ $n = 12, p = 4, N = 256$	Joint approximate diagonalization of N matrices over the set of all $n \times p$ orthonormal matrices [53].	Stiefel
P5	$n = 100, p = 30, r = 5$ $n = 500, p = 100, r = 5$ $n = 1000, p = 200, r = 5$	The robust matrix completion [3]: given some observed entries of an $n \times p$ matrix with rank r, aims to recover the complete matrix.	Fixed-rank matrices
P6	$n = 500, p = 10$ $n = 1000, p = 50$ $n = 2000, p = 100$	The closest unit norm column approximation problem [33]: given a matrix $A \in \mathbb{R}^{n \times p}$, find a matrix that is closest to A under the Frobenius norm.	Oblique
P7	$n = 5, K = 50$ $n = 10, K = 100$ $n = 20, K = 200$	Computing the Karcher Mean of $\{A_1, \ldots, A_K\}$ of $n \times n$ symmetric positive definite matrices [55].	Symmetric positive-definite matrices

The initial data for P2 are generated following the approach in [54], while those for the other problems are consistent with [56]. Regarding the manifold operations, for the Sphere, Grassmann, and Stiefel manifolds and symmetric positive definite matrices, the retraction and vector transport are given by the exponential map and parallel transport, respectively (see [29, 46, 57] for details). The corresponding operators for fixed-rank matrices follow the construction in [46, Section 7], and for the oblique manifold, they are taken from [5]. As these operators are well established in the Riemannian optimization literature, their explicit formulations are omitted here. All of these are readily available in the Manopt* toolbox for MATLAB. The initial stepsize in Algorithm 2 is set to $\alpha_k^{\mathrm{It}} = 0.5$ for all test problems. The parameters of Algorithm 2 are set to $\sigma_1 = 0.01$, $\sigma_2 = 0.999$, $z_1 = 0.875$, $z_2 = 10^{-8}$, $z_3 = 10^6$, $z_4 = 10^{-4}$, $z_5 = 10^{-8}$, $M_R = 4d$, $M_Q = 5$, and $t_k = 0.9999$. The parameter settings for all compared algorithms (Section 5.2) remain at their default values as specified in their respective references throughout the experiments. For the Riemannian Wolfe and strong Wolfe line searches used in the compared CG methods, the parameters are set to $\sigma_1 = 0.01$ and $\sigma_2 = 0.999$, consistent with standard implementations in the literature. All algorithms are terminated when the termination criterion

$$\|g_k\|_{x_k} / \|g_0\|_{x_0} \leq 10^{-6}$$

is satisfied or when the number of iterations exceeds 5000. The experimental design involves random generation of 50 test cases per problem per dimension. The random data are generated via the **randn.m** function in MATLAB. For each test problem and each dimension, all compared algorithms are initialized from the same set of starting points. All codes are implemented in Matlab R2020a and executed on a PC with an Intel (R) Core (TM) i5-1135G7 CPU (2.4 GHz) and 16 GB of RAM. The compared methods include the Riemannian Barzilai-Borwein methods RBB1 and RBB2 [58]; the Riemannian conjugate gradient methods RPRP (Riemannian Polak-Ribière-Polyak) [34], RFR (Riemannian Fletcher-Reeves) [44], RDY (Riemannian Dai-Yuan) [32], RHS (Riemannian Hestenes-Stiefel) [33], RCD (Riemannian Conjugate Descent) [30], and RLS (Riemannian Liu-Storey) [30]; and the hybrid Riemannian conjugate gradient methods RHZ [34], RHS-DY [33], and RPRP-FR [34].

5.2. Numerical results

The comparative analysis uses performance profiles [59] to evaluate algorithmic efficiency. Let $\mathcal{P}$ and $\mathcal{S}$ denote the sets of problems and solvers, respectively. For each $p \in \mathcal{P}$ and $s \in \mathcal{S}$, define $t_{p,s}$ as the number of iterations or CPU time required by solver s to solve problem p. The performance ratio $r_{p,s}$ is then given by

$$r_{p,s} = \frac{t_{p,s}}{\min\left\{t_{p,s'} : s' \in \mathcal{S}\right\}},$$

and the performance profile $\rho_s \colon \mathbb{R} \to [0, 1]$ is defined as

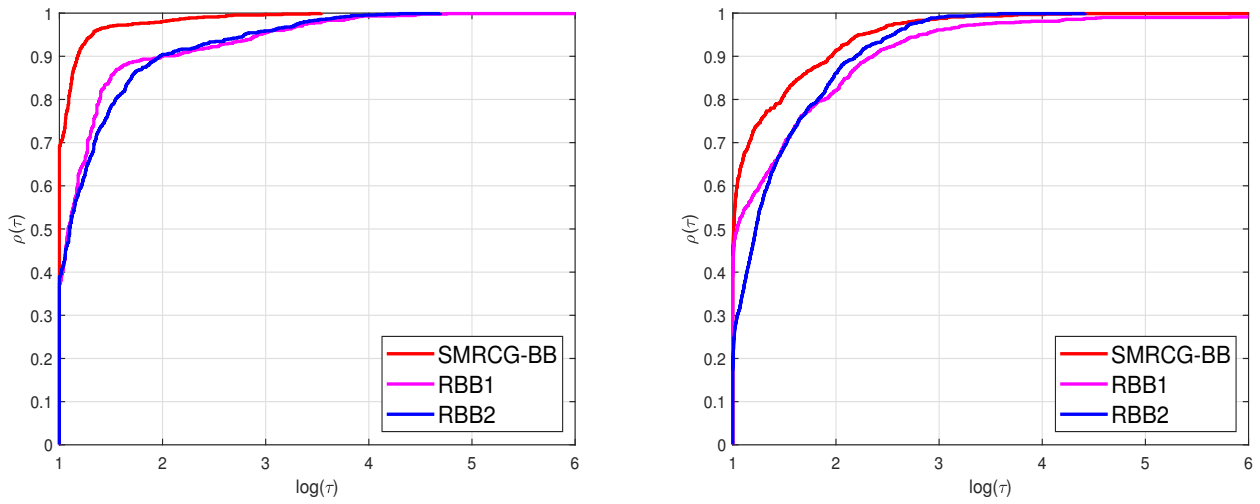
$$\rho_s(\tau) = \frac{\sharp\left\{p \in \mathcal{P} : r_{p,s} \leq \tau\right\}}{\sharp\mathcal{P}}$$

for all $\tau \in \mathbb{R}$, where $\sharp A$ denotes the number of elements of a set A. For each solver $s \in \mathcal{S}$, the performance profile ρ_s is plotted to enable comparison of algorithm performance.

*https://www.manopt.org/

The numerical comparison between SRCG-NWR and eleven competing methods is presented in Figures 1–5. In these performance profiles, curves positioned closer to the top-left region indicate superior algorithmic efficiency. The notations "·+Wolfe" and "·+strong Wolfe" distinguish implementations using Riemannian Wolfe and strong Wolfe line search conditions [44], respectively. Subfigures (a) and (b) in Figures 1–5 separately evaluate performance based on iteration counts and CPU time, providing comprehensive insights into convergence speed and computational cost.

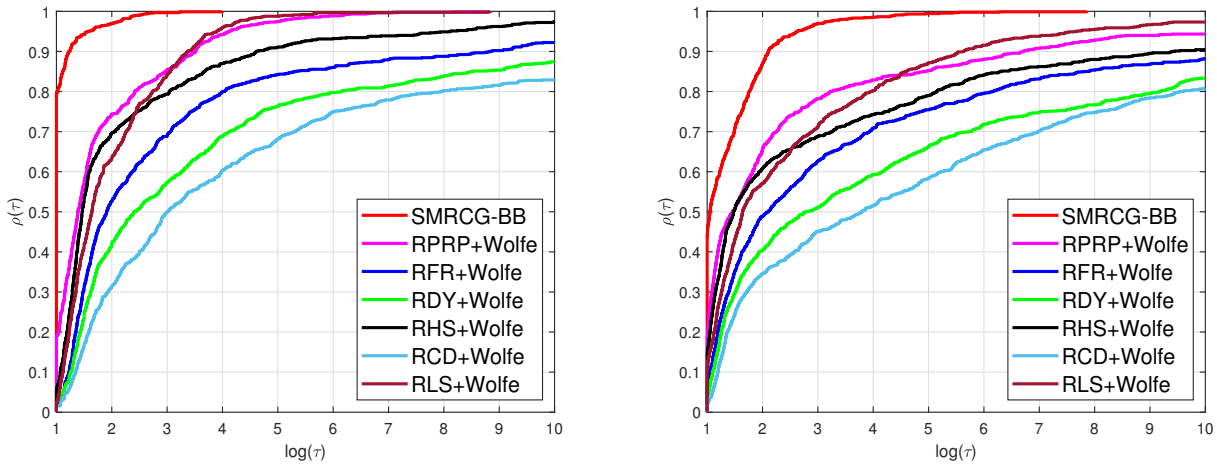
The performance profiles of SRCG-NWR and two Riemannian BB methods (RBB1 and RBB2 [58]) are presented in Figure 1. Numerical experimental results show that the SRCG-NWR method outperforms the RBB1 and RBB2 methods in both the number of iterations and CPU time. This fully validates the effectiveness of using the subspace technique for computing the search direction in the proposed method and demonstrates that this technique plays a critical role in improving the efficiency of the algorithm.



(a) Number of iterations (b) CPU time

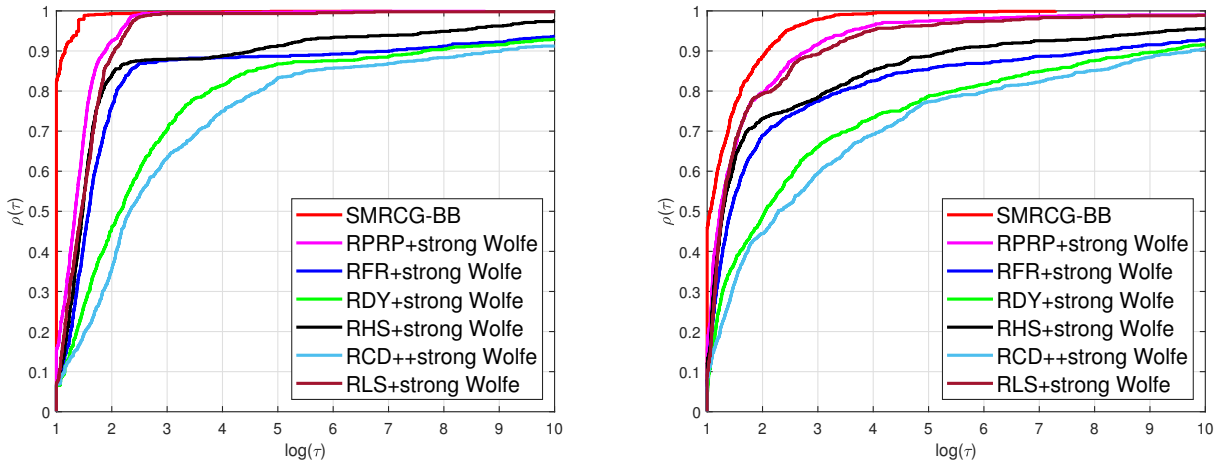
Figure 1. Performance profiles comparing SRCG-NWR with two Riemannian BB methods.

Figures 2 and 3 display the performance comparison curves between SRCG-NWR and six Riemannian CG methods (RPRP [34], RFR [44], RDY [32], RHS [33], RCD [30], and RLS [30]) under Riemannian Wolfe and strong Wolfe line search strategies, respectively. As illustrated in the figures, the proposed SRCG-NWR method demonstrates superior computational efficiency, requiring significantly fewer iterations and less CPU time compared to classical Riemannian CG methods. Numerical results also confirm the rationality and necessity of the overall strategy: constraining the search direction within the conjugate framework, adopting four calculation rules for the conjugate parameter, and using a restart criterion based on quadratic model approximation accuracy. This strategy exploits local information at different iteration stages, enabling flexible adjustment of the search direction while preserving the advantages of conjugate directions, thus achieving superior convergence.



(a) Number of iterations (b) CPU time

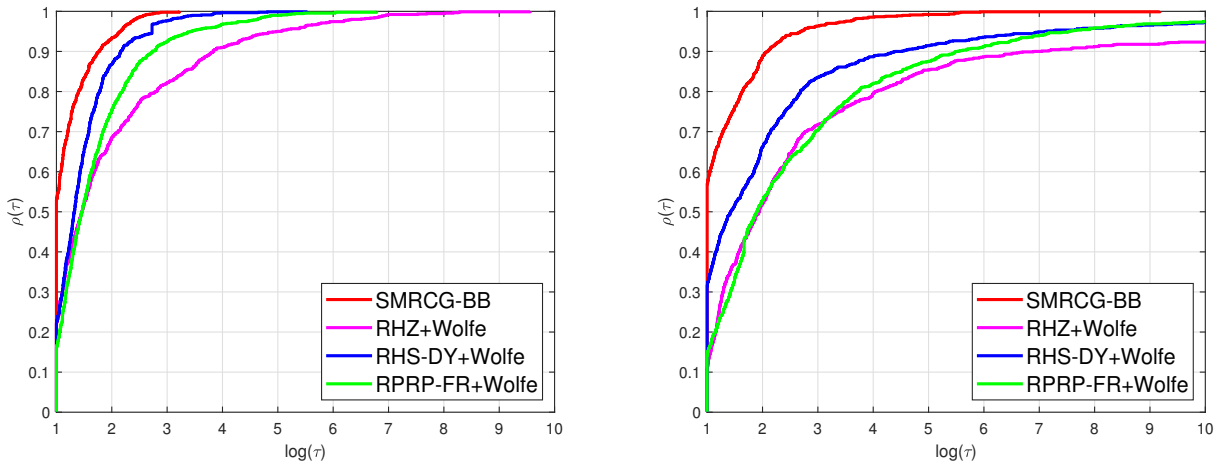
Figure 2. Performance profiles comparing SRCG-NWR with six Riemannian CG methods using Wolfe line search.



(a) Number of iterations (b) CPU time

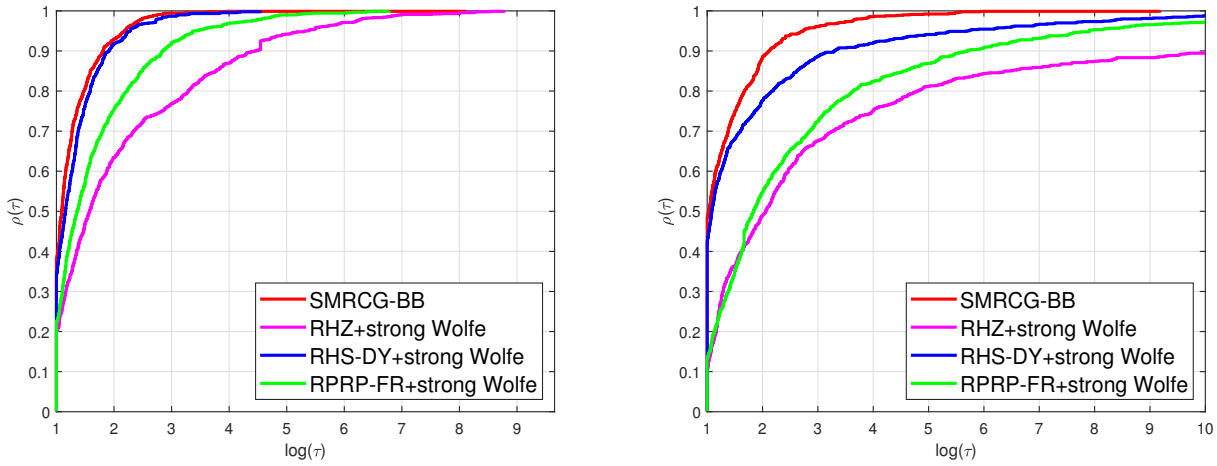
Figure 3. Performance profiles comparing SRCG-NWR with six Riemannian CG methods using strong Wolfe line search.

Figures 4 and 5 show the performance comparison curves between SRCG-NWR and three Riemannian hybrid CG methods (RHZ [34], RHS-DY [33], and RPRP-FR [34]) under the Wolfe and strong Wolfe line search strategies, respectively. As can be seen from the figures, the proposed SRCG-NWR method exhibits overall superior performance compared to the three hybrid Riemannian CG methods. Moreover, although the RHS-DY method with strong Wolfe line search achieves a similar number of iterations as the proposed SRCG-NWR method, the RHS-DY method requires more CPU time.



(a) Number of iterations (b) CPU time

Figure 4. Performance profiles comparing SRCG-NWR with three Riemannian hybrid CG methods using Wolfe line search.



(a) Number of iterations (b) CPU time

Figure 5. Performance profiles comparing SRCG-NWR with three Riemannian hybrid CG methods using strong Wolfe line search.

This is because the strong Wolfe conditions are more restrictive and typically demand more function and gradient evaluations per iteration than the nonmonotone Wolfe line search employed in our method, leading to a higher computational cost per iteration. These results again confirm the effectiveness of incorporating the subspace technique and the restart strategy into the proposed algorithm. The combination of these two strategies ensures that the method maintains stable and efficient convergence performance across various test problems, which further validates its reliability in practical applications.

Experimental results demonstrate that the proposed subspace minimization Riemannian CG method with nonmonotone Wolfe line search and a restart strategy achieves superior convergence efficiency compared to traditional Riemannian BB methods and outperforms eleven existing Riemannian CG algorithms. This advancement confirms the feasibility of integrating the nonmonotone Wolfe line search with the subspace minimization Riemannian CG framework, providing an efficient and reliable numerical method for solving Riemannian optimization problems.

6. Conclusions

This paper first proposes a novel Riemannian CG method that effectively overcomes a numerical drawback commonly found in classical approaches by incorporating a modified nonmonotone Riemannian Wolfe line search. Then, a complete algorithmic framework is established by integrating subspace minimization techniques and a restart strategy. Specifically, at each iteration, the search direction is computed by minimizing a quadratic approximation of the objective function over a two-dimensional subspace, which admits a closed-form expression. The global convergence and the R-linear convergence rate of the proposed algorithm are rigorously established under mild assumptions. Numerical experiments demonstrate that the proposed method outperforms existing Riemannian BB methods, six Riemannian CG methods, and three hybrid Riemannian CG methods in terms of both iteration counts and CPU time.

Author contributions

Shajie Xing: wrote the original draft and developed the software; Huangyue Chen: designed the methodology and acquired the funding; Chunming Tang: contributed to funding acquisition and writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

During the preparation of this manuscript, the authors used Deepseek for language polishing and grammar checking. After using this tool, the authors reviewed and revised the content and assume full responsibility for the final version of the article.

Acknowledgments

The authors sincerely thank the Editor and reviewers for their constructive comments, which have greatly improved this manuscript. This work was supported by the Guangxi Natural Science Foundation (2026GXNSFDA00640024, 2026GXNSFBA00640129), the National Natural Science Foundation of China (12271113, 12501452), the Special Foundation for Guangxi Ba Gui Scholars (GXR-6BG2424008) and the Middle-aged and Young Teachers' Basic Ability Promotion Project of Guangxi (2025KY0025).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Y. Saad, *Numerical methods for large eigenvalue problems*, SIAM, 2011. https://doi.org/10.1137/1.9781611970739
2. Z. Wen, C. Yang, X. Liu, Y. Zhang, Trace-penalty minimization for large-scale eigenspace computation, *J. Sci. Comput.*, **66** (2016), 1175–1203. https://doi.org/10.1007/s10915-015-0061-0
3. B. Vandereycken, Low-rank matrix completion by Riemannian optimization, *SIAM J. Optim.*, **23** (2013), 1214–1236. https://doi.org/10.1137/110845768
4. D. Kressner, M. Steinlechner, B. Vandereycken, Low-rank tensor completion by Riemannian optimization, *BIT Numer. Math.*, **54** (2014), 447–468. https://doi.org/10.1007/s10543-013-0455-z
5. P. A. Absil, K. A. Gallivan, Joint diagonalization on the oblique manifold for independent component analysis, *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2014. https://doi.org/10.1109/ICASSP.2006.1661433
6. P. Comon, G. H. Golub, Tracking a few extreme singular values and vectors in signal processing, *Proc. IEEE*, **78** (1990), 1327–1343. https://doi.org/10.1109/5.58320
7. P. H. Schönemann, A generalized solution of the orthogonal procrustes problem, *Psychometrika*, **31** (1966), 1–10. https://doi.org/10.1007/BF02289451
8. L. Eldén, H. Park, A procrustes problem on the Stiefel manifold, *Numer. Math.*, **82** (1999), 599–619. https://doi.org/10.1007/s002110050432
9. M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand.*, **49** (1952), 409–436. https://doi.org/10.6028/JRES.049.044
10. R. Fletcher, C. M. Reeves, Function minimization by conjugate gradients, *Comput. J.*, **7** (1964), 149–154. https://doi.org/10.1093/comjnl/7.2.149
11. E. Polak, G. Ribiere, Note sur la convergence de méthodes de directions conjuguées, *Rev. Fr. Inf. Rech. Opér. Sér. Rouge*, **3** (1969), 35–43. https://doi.org/10.1051/m2an/196903R100351
12. B. T. Polak, The conjugate gradient method in extremal problems, *USSR Comput. Math. Math. Phys.*, **9** (1969), 94–112. https://doi.org/10.1016/0041-5553(69)90035-4
13. Y. H. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optim.*, **10** (1999), 177–182. https://doi.org/10.1137/S1052623497318992
14. Y. H. Dai, Y. Yuan, An efficient hybrid conjugate gradient method for unconstrained optimization, *Ann. Oper. Res.*, **103** (2001), 33–47. https://doi.org/10.1023/A:1012930416777

15. L. Zhang, W. Zhou, D. H. Li, A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence, *IMA J. Numer. Anal.*, **26** (2006), 629–640. https://doi.org/10.1093/imanum/drl016

16. Z. Wan, Z. Yang, Y. Wang, New spectral PRP conjugate gradient method for unconstrained optimization, *Appl. Math. Lett.*, **24** (2011), 16–22. https://doi.org/10.1016/j.aml.2010.08.002

17. Y. Xiao, H. Song, Z. Wang, A modified conjugate gradient algorithm with cyclic Barzilai–Borwein steplength for unconstrained optimization, *J. Comput. Appl. Math.*, **236** (2012), 3101–3110. https://doi.org/10.1016/j.cam.2012.01.032

18. J. Liu, Y. Feng, L. Zou, A spectral conjugate gradient method for solving large-scale unconstrained optimization, *Comput. Math. Appl.*, **77** (2019), 731–739. https://doi.org/10.1016/j.camwa.2018.10.002

19. X. Jiang, H. Yang, J. Yin, W. Liao, A three-term conjugate gradient algorithm with restart procedure to solve image restoration problems, *J. Comput. Appl. Math.*, **424** (2023), 115020. https://doi.org/10.1016/j.cam.2022.115020

20. Y. X. Yuan, J. Stoer, A subspace study on conjugate gradient algorithms, *Z. Angew. Math. Mech.*, **75** (1995), 69–77. https://doi.org/10.1002/zamm.19950750118

21. N. Andrei, An accelerated subspace minimization three-term conjugate gradient algorithm for unconstrained optimization, *Numer. Algorithms*, **65** (2014), 859–874. https://doi.org/10.1007/s11075-013-9718-7

22. Y. Dai, C. Kou, A Barzilai-Borwein conjugate gradient method, *Sci. China Math.*, **59** (2016), 1511–1524. https://doi.org/10.1007/s11425-016-0279-2

23. J. Barzilai, J. M. Borwein, Two-point step size gradient methods, *IMA J. Numer. Anal.*, **8** (1988), 141–148. https://doi.org/10.1093/imanum/8.1.141

24. M. Li, H. Liu, Z. Liu, A new subspace minimization conjugate gradient method with nonmonotone line search for unconstrained optimization, *Numer. Algorithms*, **79** (2018), 195–219. https://doi.org/10.1007/s11075-017-0434-6

25. H. Liu, Z. Liu, An efficient Barzilai–Borwein conjugate gradient method for unconstrained optimization, *J. Optim. Theory Appl.*, **180** (2019), 879–906. https://doi.org/10.1007/s10957-018-1393-3

26. Z. Liu, Y. Ni, H. Liu, W. Sun, A new subspace minimization conjugate gradient method for unconstrained minimization, *J. Optim. Theory Appl.*, **200** (2024), 820–851. https://doi.org/10.1007/s10957-023-02325-x

27. A. Lichnewsky, Une methode de gradient conjugue sur des varietes application a certains problemes de valeurs propres non lineaires, *Numer. Funct. Anal. Optim.*, **1** (1979), 515–560. https://doi.org/10.1080/01630567908816032

28. S. T. Smith, Optimization techniques on Riemannian manifolds, *Fields Inst. Commun.*, 1994.

29. P. A. Absil, R. Mahony, R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2009. https://doi.org/10.1515/9781400830244

30. H. Sato, Riemannian conjugate gradient methods: general framework and specific algorithms with convergence analyses, *SIAM J. Optim.*, **32** (2022), 2690–2717. https://doi.org/10.1137/21M1464178

31. H. Sato, T. Iwai, A new, globally convergent Riemannian conjugate gradient method, *Optimization*, **64** (2015), 1011–1031. https://doi.org/10.1080/02331934.2013.836650

32. H. Sato, A Dai–Yuan-type Riemannian conjugate gradient method with the weak Wolfe conditions, *Comput. Optim. Appl.*, **64** (2016), 101–118. https://doi.org/10.1007/s10589-015-9801-1

33. H. Sakai, H. Iiduka, Hybrid Riemannian conjugate gradient methods with global convergence properties, *Comput. Optim. Appl.*, **77** (2020), 811–830. https://doi.org/10.1007/s10589-020-00224-9

34. H. Sakai, H. Iiduka, Sufficient descent Riemannian conjugate gradient methods, *J. Optim. Theory Appl.*, **190** (2021), 130–150. https://doi.org/10.1007/s10957-021-01874-3

35. C. Tang, X. Rong, J. Jian, S. Xing, A hybrid Riemannian conjugate gradient method for nonconvex optimization problems, *J. Appl. Math. Comput.*, **69** (2023), 823–852. https://doi.org/10.1007/s12190-022-01772-5

36. N. Salihu, P. Kumam, S. Salisu, K. Khammahawong, Some hybrid Riemannian conjugate gradient methods with restart strategy, *Franklin Open*, **10** (2025), 100240. https://doi.org/10.1016/j.fraope.2025.100240

37. C. Tang, W. Tan, S. Xing, H. Zheng, A class of spectral conjugate gradient methods for Riemannian optimization, *Numer. Algorithms*, **94** (2023), 131–147. https://doi.org/10.1007/s11075-022-01495-5

38. C. Tang, W. Tan, Y. Zhang, Z. Liu, An accelerated spectral CG based algorithm for optimization techniques on Riemannian manifolds and its comparative evaluation, *J. Comput. Appl. Math.*, **462** (2025), 116482. https://doi.org/10.1016/j.cam.2024.116482

39. S. Nasiru, P. Kumam, S. Salisu, L. Wang, T. Seangwattana, Spectral conjugate gradient for Riemannian optimization: application to the Gough–Stewart platform, *Int. J. Dyn. Control*, **13** (2025), 288. https://doi.org/10.1007/s40435-025-01788-2

40. Y. Wang, H. Shao, H. Sun, T. Wu, An adaptive restart Riemannian spectral Dai-Kou conjugate gradient method with quasi-Newton update, *J. Appl. Math. Comput.*, **72** (2026), 128. https://doi.org/10.1007/s12190-026-02773-4

41. N. Salihu, P. Kumam, S. Salisu, K. Sitthithakerngkiet, On new spectral conjugate gradient methods for Riemannian optimization using retraction and scaled vector transport, *Oper. Res. Forum*, **7** (2026), 7. https://doi.org/10.1007/s43069-025-00594-y

42. N. Salihu, P. Kumam, L. Wang, S. Salisu, Some new three-term conjugate gradient methods for Riemannian optimization with application to the Gough-Stewart platform, *Math. Model. Numer. Simul. Appl.*, **5** (2025), 2. https://doi.org/10.53391/2791-8564.1001

43. Y. Wang, H. Shao, H. Sun, M. Jiang, The global convergence of a spectral three-term Riemannian conjugate gradient method with convex combination, *Oper. Res. Lett.*, **65** (2026), 107406. https://doi.org/10.1016/j.orl.2026.107406

44. W. Ring, B. Wirth, Optimization methods on Riemannian manifolds and their application to shape space, *SIAM J. Optim.*, **22** (2012), 596–627. https://doi.org/10.1137/11082885x

45. H. Sato, *Riemannian optimization and its applications*, Springer, 2021. https://doi.org/10.1007/978-3-030-62391-3

46. N. Boumal, *An introduction to optimization on smooth manifolds*, Cambridge University Press, 2023. https://doi.org/10.1017/9781009166164

47. Y. H. Dai, C. X. Kou, A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search, *SIAM J. Optim.*, **23** (2013), 296–320. https://doi.org/10.1137/100813026

48. H. Zhang, W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM J. Optim.*, **14** (2004), 1043–1056. https://doi.org/10.1137/S1052623403428208

49. Y. Narushima, S. Nakayama, M. Takemura, H. Yabe, Memoryless quasi-Newton methods based on the spectral-scaling Broyden family for Riemannian optimization, *J. Optim. Theory Appl.*, **197** (2023), 639–664. https://doi.org/10.1007/s10957-023-02183-7

50. W. Huang, K. A. Gallivan, P. A. Absil, A Broyden class of quasi-Newton methods for Riemannian optimization, *SIAM J. Optim.*, **25** (2015), 1660–1685. https://doi.org/10.1137/140955483

51. W. Huang, P. A. Absil, K. A. Gallivan, A Riemannian BFGS method without differentiated retraction for nonconvex optimization problems, *SIAM J. Optim.*, **28** (2018), 470–495. https://doi.org/10.1137/17M1127582

52. X. Li, X. Wang, M. K. Lal, A nonmonotone trust region method for unconstrained optimization problems on Riemannian manifolds, *J. Optim. Theory Appl.*, **188** (2021), 547–570. https://doi.org/10.1007/s10957-020-01796-6

53. W. Huang, P. A. Absil, K. A. Gallivan, A Riemannian symmetric rank-one trust-region method, *Math. Program.*, **150** (2015), 179–216. https://doi.org/10.1007/s10107-014-0765-1

54. W. Huang, *Optimization algorithms on Riemannian manifolds with applications*, Ph.D. Thesis, The Florida State University, 2013.

55. M. Moakher, A differential geometric approach to the geometric mean of symmetric positive-definite matrices, *SIAM J. Matrix Anal. Appl.*, **26** (2005), 735–747. https://doi.org/10.1137/S0895479803436937

56. S. Xing, H. Chen, C. Tang, A class of generalized Barzilai-Borwein methods for Riemannian optimization, *Appl. Numer. Math.*, **224** (2026), 106–118. https://doi.org/10.1016/j.apnum.2026.01.019

57. R. Ferreira, J. Xavier, J. P. Costeira, V. Barroso, Newton method for Riemannian centroid computation in naturally reductive homogeneous spaces, *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006. https://doi.org/10.1109/ICASSP.2006.1660751

58. B. Iannazzo, M. Porcelli, The Riemannian Barzilai–Borwein method with nonmonotone line search and the matrix geometric mean computation, *IMA J. Numer. Anal.*, **38** (2018), 495–517. https://doi.org/10.1093/imanum/drx015

59. E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2002), 201–213. https://doi.org/10.1007/s101070100263

Appendix

Numerical results

Tables A.1 and A.2 present a detailed numerical comparison of the proposed method, two Riemannian BB methods, and nine Riemannian conjugate gradient methods on seven test problems. Each problem is run 50 times independently with random initialization for each dimension, and the values in the table are the average of the 50 experimental results. In the table, "·+Wolfe" and "·+sWolfe" indicate implementations using the Riemannian Wolfe and strong Wolfe line search conditions, respectively.

Table A.1. Comparison results of all methods for P1–P4.

Prob.	P1						P2						P3						P4					
	n						(n,p)						(n,p)						(n,p,N)					
	100		500		1000		(50, 5)		(100, 10)		(200, 20)		(100, 5)		(200, 10)		(500, 20)		(12, 4, 16)		(12, 4, 64)		(12, 4, 256)	
	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time
SRCG-NWR	**91**	**0.007**	201	**0.023**	260	**0.051**	**84**	**0.008**	**106**	**0.040**	**164**	**0.203**	275	0.067	**615**	**0.248**	**1521**	**3.619**	**99**	**0.024**	**96**	**0.037**	**94**	**0.124**
RBB1	125	0.009	217	0.018	274	0.074	101	0.013	137	0.041	183	0.257	331	0.086	758	0.300	1613	4.280	211	0.065	232	0.099	223	0.342
RBB2	102	0.008	261	0.025	280	0.064	108	0.013	144	0.045	189	0.260	310	0.061	747	0.295	1647	4.358	197	0.061	228	0.090	230	0.355
RPRP+Wolfe	126	0.012	284	0.040	367	0.111	139	0.017	233	0.074	264	0.478	522	0.077	981	0.403	1897	4.892	249	0.072	287	0.162	283	0.471
RPRP+ sWolfe	131	0.012	234	0.037	277	0.060	107	0.014	164	0.044	183	0.228	342	0.080	683	0.313	1806	4.797	201	0.066	254	0.162	262	0.465
RFR+Wolfe	284	0.022	658	0.066	1032	0.242	141	0.017	218	0.068	338	0.499	690	0.203	1178	0.662	2142	6.334	288	0.081	229	0.185	230	0.436
RFR+ sWolfe	255	0.020	442	0.054	606	0.182	111	0.013	162	0.042	236	0.266	410	0.093	956	0.357	2355	6.588	204	0.071	207	0.188	222	0.430
RDY+Wolfe	798	0.054	1955	0.241	1549	0.590	140	0.016	228	0.067	367	0.579	1022	0.297	2315	1.149	3797	7.764	256	0.078	243	0.153	224	0.428
RDY+ sWolfe	654	0.043	1018	0.125	991	0.249	107	0.013	165	0.042	261	0.318	734	0.165	1987	0.752	3627	7.145	224	0.056	234	0.169	217	0.422
RHS+Wolfe	126	0.011	276	0.027	395	0.099	126	0.017	286	0.160	313	0.495	572	0.171	899	0.443	1887	4.178	250	0.079	228	0.142	206	0.381
RHS+ sWolfe	113	0.009	331	0.038	400	0.110	101	0.013	187	0.039	220	0.254	405	0.087	857	0.304	2006	4.443	212	0.062	196	0.103	204	0.379
RCD+Wolfe	816	0.055	2473	0.337	2408	1.387	157	0.018	269	0.079	470	0.693	1222	0.357	3287	1.606	4992	10.785	297	0.088	268	0.165	264	0.497
RCD+ sWolfe	577	0.039	1214	0.127	1204	0.289	121	0.014	189	0.048	305	0.346	1101	0.228	3398	1.219	5000	7.571	227	0.057	249	0.172	254	0.488
RLS+Wolfe	163	0.013	336	0.040	321	0.080	144	0.018	212	0.068	314	0.514	748	0.248	963	0.310	1959	5.175	251	0.077	262	0.165	234	0.455
RLS+ sWolfe	116	0.010	247	0.027	335	0.089	113	0.014	155	0.043	214	0.268	417	0.091	786	0.286	1874	4.120	198	0.065	241	0.157	228	0.437
RHZ+Wolfe	127	0.013	234	0.059	273	0.068	110	0.015	192	0.058	250	0.425	522	0.098	814	0.356	3201	6.772	276	0.068	287	0.174	284	0.497
RHZ+ sWolfe	127	0.013	211	0.050	263	0.065	107	0.016	164	0.059	204	0.414	528	0.095	790	0.375	2500	6.132	288	0.099	234	0.152	234	0.478
RHS-DY+Wolfe	118	0.011	206	0.030	266	0.064	107	0.014	169	0.054	184	0.314	270	0.076	690	0.326	2359	6.434	117	0.094	115	0.099	124	0.222
RHS-DY+ sWolfe	98	0.008	**199**	0.030	**259**	0.061	106	0.014	169	0.050	167	0.233	**267**	**0.066**	625	0.273	1695	4.051	107	0.058	112	0.098	119	0.204
RPRP-FR+Wolfe	126	0.013	234	0.045	274	0.070	108	0.018	204	0.059	208	0.355	329	0.080	854	0.476	2110	6.293	230	4.332	235	0.172	228	0.467
RPRP-FR+ sWolfe	118	0.018	220	0.044	268	0.067	104	0.022	188	0.083	196	0.324	303	0.077	824	0.347	1807	4.332	216	0.104	225	0.158	216	0.451

Table A.2. Comparison results of all methods for P5–P7.

Prob.	P5						P6						P7					
	(n, p, r)						(n, p)						(n, K)					
	(100, 30, 5)		(500, 100, 5)		(1000, 200, 5)		(500, 10)		(1000, 50)		(2000, 100)		(5, 50)		(10, 100)		(20, 200)	
	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter	time
SRCG-NWR	20	0.007	**50**	**0.066**	**64**	**0.255**	**18**	**0.006**	**12**	**0.011**	**11**	**0.075**	**4**	**0.019**	**4**	**0.054**	**4**	**0.188**
RBB1	**17**	**0.006**	52	0.069	68	0.259	**18**	0.007	15	0.016	14	0.097	**4**	0.021	4	**0.054**	**4**	**0.188**
RBB2	**17**	**0.006**	53	0.080	69	0.304	19	0.008	16	0.014	14	0.088	**4**	0.022	**4**	**0.054**	**4**	**0.188**
RPRP+Wolfe	22	0.014	120	0.132	211	0.956	23	0.009	19	0.029	16	0.128	7	0.038	7	0.112	8	0.493
RPRP+ sWolfe	22	0.008	80	0.116	136	0.614	24	0.014	14	0.017	12	0.093	7	0.040	7	0.110	7	0.406
RFR+Wolfe	42	0.015	461	0.655	1429	6.251	27	0.011	18	0.030	16	0.146	7	0.039	7	0.105	7	0.322
RFR+ sWolfe	41	0.016	444	0.623	1415	6.880	25	0.012	15	0.017	13	0.098	7	0.042	7	0.113	7	0.342
RDY+Wolfe	41	0.015	454	0.648	1448	6.322	27	0.011	18	0.031	16	0.146	7	0.038	7	0.100	7	0.472
RDY+ sWolfe	45	0.018	465	0.650	1406	6.879	25	0.011	16	0.018	13	0.087	7	0.041	7	0.111	8	0.425
RHS+Wolfe	29	0.010	270	0.380	759	3.254	29	0.013	19	0.035	16	0.159	7	0.036	6	0.090	6	0.331
RHS+ sWolfe	29	0.012	270	0.378	763	3.515	26	0.013	15	0.018	**11**	0.088	7	0.037	7	0.092	7	0.383
RCD+Wolfe	57	0.020	531	0.696	1489	6.278	27	0.011	18	0.031	16	0.142	7	0.040	7	0.128	9	1.134
RCD+ sWolfe	56	0.022	514	0.654	1459	6.981	23	0.011	15	0.016	12	0.092	7	0.039	8	0.114	7	0.322
RLS+Wolfe	22	0.008	87	0.115	146	0.590	28	0.013	19	0.037	17	0.161	6	0.032	7	0.095	8	0.453
RLS+ sWolfe	22	0.010	95	0.119	101	0.344	24	0.011	15	0.018	13	0.088	7	0.037	7	0.092	7	0.376
RHZ+Wolfe	23	0.007	119	0.198	146	0.661	24	0.010	18	0.020	16	0.111	7	0.038	8	0.113	7	0.354
RHZ+ sWolfe	22	0.009	111	0.180	159	0.647	23	0.011	18	0.018	13	0.089	6	0.032	7	0.104	7	0.353
RHS-DY+Wolfe	24	0.009	57	0.094	80	0.351	25	0.011	16	0.019	14	0.093	5	0.027	6	0.084	6	0.324
RHS-DY+ sWolfe	22	0.007	52	0.089	70	0.298	25	0.013	17	0.018	13	0.089	5	0.027	6	0.085	6	0.322
RPRP-FR+Wolfe	22	0.008	55	0.096	101	0.333	25	0.009	18	0.021	15	0.106	5	0.027	6	0.083	6	0.333
RPRP-FR+ sWolfe	21	0.009	54	0.099	87	0.324	29	0.009	16	0.017	15	0.097	7	0.028	6	0.085	6	0.323