Journal of Industrial and Management Optimization

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

JIMO, 22(12): 6539–6558.
DOI:10.3934/jimo.2026219
Received: 28 September 2026
Revised: 28 October 2026
Accepted: 10 November 2026
Published: 17 November 2026

Research article

Barzilai–Borwein stepsizes with nonmonotone line search for proximal gradient methods on the Stiefel manifold

Di Tang[1] **and Lin Chen**[2,*]

[1] School of Mathematical Sciences, Chongqing Normal University, Chongqing 401331, China
[2] National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing 401331, China

* **Correspondence:** Email: allenchen@cqnu.edu.cn.

Abstract: ManPG for nonsmooth optimization on the Stiefel manifold uses a fixed proximal parameter $t = 1/L$. When the global Lipschitz constant overestimates local curvature, steps become far too small; a moderately larger fixed t already recovers most of the speed. We replace t by Barzilai–Borwein (BB) stepsizes computed from ambient secants, with the absolute curvature $|\langle s, y\rangle|$ so that the rule stays active on concave quadratics where the signed guard freezes at $1/L$. Because BB values oscillate naturally, a monotone Armijo test would backtrack excessively; we therefore pair the adaptive parameter with a GLL nonmonotone line search on the retraction stepsize, which tolerates occasional objective increases over a sliding window. The analysis covers any safeguarded $t_k \in [1/L, t_{\max}]$ for exact subproblem solves: $\|V_k\|_F \to 0$, every accumulation point is stationary, and $O(\epsilon^{-2})$ complexity holds. The main contributions are (i) a matched-start ablation showing that $t = 1/L$ is the main bottleneck on ill-conditioned problems; (ii) a safeguarded alternating BB rule for the proximal parameter with the absolute-curvature policy, together with global convergence and $O(\epsilon^{-2})$ complexity; and (iii) a matched protocol with comparisons against ManPG, ManPG-Ada, and scalar reimplementations of an outer-stepsize-BB nonmonotone proximal method (NPGL-type) and of a scalar-metric proximal quasi-Newton scheme. On compressed modes and ℓ_1-regularized PCA, BB-ManPG converges reliably where fixed-parameter ManPG is slow or fails, and ablations separate the contributions of the adaptive parameter, the nonmonotone criterion, and the warm start.

Keywords: Stiefel manifold; proximal gradient method; Barzilai–Borwein stepsize; nonmonotone line search; nonsmooth optimization
Mathematics Subject Classification: 65K05, 90C26, 90C30, 90C56

1. Introduction

Problems over the Stiefel manifold $\mathrm{St}(n,r) = \{X \in \mathbb{R}^{n\times r} : X^\top X = I_r\}$ arise across machine learning and scientific computing. Compressed modes of the Schrödinger equation [1], sparse principal component analysis [2], unsupervised feature selection [3], and sparse blind deconvolution [4] each reduce to a composite objective

$$\min_{X\in\mathrm{St}(n,r)} \; F(X) := f(X) + h(X), \tag{1}$$

where f is continuously differentiable with L-Lipschitz-continuous gradient (possibly nonconvex) and h is finite-valued convex on $\mathbb{R}^{n\times r}$ and typically nonsmooth. These applications fit (1) with concrete choices of f and h: For compressed modes, $f(X) = \mathrm{Tr}(X^\top HX)$ with H a discretized Hamiltonian and $h(X) = \mu\|X\|_1$; for sparse PCA, $f(X) = -\mathrm{Tr}(X^\top MX)$ with a data covariance-type matrix M and $h(X) = \mu\|X\|_1$; for unsupervised feature selection one typically takes an $\ell_{2,1}$ penalty as h; and sparse blind deconvolution leads to a sphere-constrained sparse recovery problem of the same composite form. The ℓ_1 penalty $\|X\|_1 = \sum_{ij}|X_{ij}|$ is the leading example of h.

Smooth Riemannian optimization is well developed [5–7]. For nonsmooth composite objectives one has subgradient and stochastic projected-gradient methods [8–10], proximal point methods [11–13], ADMM splittings—the splitting method of [14] for orthogonality-constrained problems, the augmented Lagrangian method for ℓ_1-regularized problems with orthogonality constraints of [15], the generic manifold ADMM algorithm MADMM of [16], and the inexact manifold augmented Lagrangian method of [17]—and Riemannian proximal gradients [18,19]. Penalty and constraint-dissolving techniques cover orthogonality constraints: The new first-order algorithmic framework of [20], the exact penalty function for $\ell_{2,1}$-norm minimization of [21], the constraint dissolving approach for Riemannian optimization of [22], the constraint dissolving approach for nonsmooth problems over the Stiefel manifold of [23], the penalty-free infeasible approach of [24], and the smooth exact penalty function of [25]. Some of these approaches require penalty tuning that is difficult in practice [26]; others come with global guarantees only under additional structural assumptions [27]. Riemannian proximal gradient schemes come with their own convergence theory [18]. The manifold proximal gradient method (ManPG) of Chen, Ma, So, and Zhang [26] fixes these issues. At each iteration, ManPG builds a quadratic model of f on the tangent space and solves a strongly convex proximal subproblem, then pulls the result back to the manifold via retraction. The method converges globally to stationary points with $O(\epsilon^{-2})$ iteration complexity. The inner subproblem is solved by a semismooth Newton method applied to the dual, which serves as the core computational routine of the algorithm.

ManPG, like its Euclidean counterpart, uses a fixed proximal parameter $t = 1/L$ where L is the global Lipschitz constant of ∇f (its convergence theory allows any fixed $t > 0$; $1/L$ is the conventional default). The choice works in theory but causes trouble in practice. On the compressed modes problem, L grows quadratically with the discretization dimension, forcing t to be correspondingly tiny and slowing convergence substantially. When the local curvature of f is mild, the global constant badly underestimates the viable step size. The primary bottleneck is the mismatch between the Lipschitz constant and the local curvature, not the ManPG framework itself. Simply replacing $t = 1/L$ with a moderately larger fixed constant (say $t = 10/L$) already recovers most of the lost performance. Chen et al. [26] recognized the issue (their Section 6.1 observes that enlarging t can reduce outer iterations) and proposed ManPG-Ada, a heuristic that changes t multiplicatively depending on whether backtracking

occurred. The heuristic helps, but it uses only binary feedback and introduces a tuning parameter τ whose best value varies across problems.

The Barzilai–Borwein (BB) method [28] approximates local curvature from successive iterates at negligible cost. SpaRSA [29] already pairs BB with a nonmonotone line search [30] in Euclidean composite optimization; Riemannian BB variants target smooth problems [31, 32].

The closest prior art is as follows. ManPQN [33] uses a variable metric in the same tangent model with GLL search (our $t_k^{-1}I$ is a scalar specialization); NPGL [34] uses BB on the *outer* step with fixed proximal t. Our contribution is the safeguarded BB rule on the *proximal parameter* (with absolute curvature so it stays active on concave quadratics), exact-solve theory for any bounded t_k, and matched-start diagnostics.

Table 1 and the experiments below position BB-ManPG relative to ManPG, ManPG-Ada, the NPGL-type outer-BB nonmonotone method, and the PQN-s scalar-metric scheme.

Extending BB to the manifold proximal gradient setting takes some care, for three reasons. First, the ManPG subproblem differs structurally from a standard proximal gradient step: It lives on the tangent space, and the proximal parameter t controls the quadratic penalty rather than directly scaling the step. Second, existing Riemannian spectral and quasi-Newton methods [31, 35] target smooth problems and adapt the stepsize along a Riemannian gradient direction; by contrast, ManPG involves a proximal subproblem whose solution V_k is not simply a scaled gradient. Third, on problem classes where the smooth term is a concave quadratic on the sphere (e.g. ℓ_1-regularized PCA), the ambient secant satisfies $\langle s_k, y_k\rangle < 0$ at every step. Any BB rule that guards on the sign of $\langle s_k, y_k\rangle$ then never fires, and the parameter stays frozen at $1/L$. We address the first two points by adapting the BB rule to t_k through ambient-space differences, and the third by taking the absolute curvature $|\langle s_k, y_k\rangle|$ in the BB denominator, so that the rule estimates the magnitude of local curvature regardless of its sign. The acceptance test must also be chosen with care: Since BB values oscillate naturally, a monotone Armijo condition would trigger excessive backtracking. Pairing the rule with the GLL nonmonotone line search on the retraction stepsize α addresses this issue and preserves the $O(\epsilon^{-2})$ complexity guarantee of ManPG; the residuals converge to zero for every fixed window length M, so every accumulation point is stationary.

The contributions of this paper are as follows: (i) A matched-start ablation showing $t = 1/L$ is the main bottleneck on ill-conditioned problems; (ii) An alternating BB rule for t_k with $|\langle s, y\rangle|$, GLL line search, global convergence and $O(\epsilon^{-2})$ complexity for exact solves; (iii) A matched protocol and comparisons with ManPG, ManPG-Ada, NPGL-type and scalar-metric baselines.

Section 2 collects the necessary background on the Stiefel manifold and the ManPG algorithm. Section 3 describes the BB adaptation, the absolute-curvature policy, and the nonmonotone line search integration. Convergence and complexity are treated in Section 4. Section 5 reports numerical experiments under a common controlled protocol, including the ablations; Section 6 closes with a summary and limitations.

2. Preliminaries

Let $\mathcal{M} = \mathrm{St}(n, r) = \{X \in \mathbb{R}^{n\times r} : X^\top X = I_r\}$ with the Frobenius inner product. The tangent space is $T_X\mathcal{M} = \{V : V^\top X + X^\top V = 0\}$ and the orthogonal projector is

$$\mathrm{Proj}_{T_X\mathcal{M}}(Y) = (I_n - XX^\top)Y + \tfrac{1}{2}X(X^\top Y - Y^\top X). \tag{2}$$

For smooth f, $\mathrm{grad} f(X) = \mathrm{Proj}_{T_X\mathcal{M}}(\nabla f(X))$. A retraction Retr_X satisfies $\mathrm{Retr}_X(0) = X$, $\mathrm{DRetr}_X(0) = \mathrm{id}$, and on a compact $\mathcal{M}$,

$$\|\mathrm{Retr}_X(\xi) - X\|_F \le M_1\|\xi\|_F, \qquad \|\mathrm{Retr}_X(\xi) - (X+\xi)\|_F \le M_2\|\xi\|_F^2 \tag{3}$$

for $\|\xi\|_F$ small; we use the polar retraction $(X+\xi)(I+\xi^\top\xi)^{-1/2}$ (implemented via a thin SVD of $X+\xi$).

For $F = f + h$ with f smooth and h finite convex, $X \in \mathcal{M}$ is *stationary* if

$$0 \in \mathrm{grad} f(X) + \mathrm{Proj}_{T_X\mathcal{M}}(\partial h(X)). \tag{4}$$

ManPG [26] solves the tangent subproblem

$$V_t(X) = \arg\min_{V \in T_X\mathcal{M}} \langle \nabla f(X), V\rangle + \frac{1}{2t}\|V\|_F^2 + h(X+V), \tag{5}$$

which is uniquely defined for every $t > 0$, and sets $X^+ = \mathrm{Retr}_X(\alpha V)$ with Armijo α. With $t = 1/L$ every limit point is stationary and an ϵ-stationary point is reached in $O(\epsilon^{-2})$ iterations; the theory allows any fixed $t > 0$, and $1/L$ is the conventional default.

3. The BB-ManPG algorithm

We now describe the proposed method. The core idea: Replace the fixed proximal parameter t with an adaptively chosen t_k updated by the Barzilai–Borwein rule, paired with a nonmonotone line search to handle the oscillatory stepsizes that BB methods naturally produce.

3.1. Barzilai–Borwein stepsizes in the ambient space

For unconstrained smooth minimization $\min_x f(x)$, the BB stepsize [28] uses

$$\alpha_k^{\mathrm{BB1}} = \frac{\|s_{k-1}\|_F^2}{\langle s_{k-1}, y_{k-1}\rangle}, \qquad \alpha_k^{\mathrm{BB2}} = \frac{\langle s_{k-1}, y_{k-1}\rangle}{\|y_{k-1}\|_F^2}, \tag{6}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$. For a quadratic f both formulas return $1/\lambda$ on an eigenvector of $\nabla^2 f$; in general BB1 is a long step and BB2 a short step [36]. We alternate BB1 (even k) and BB2 (odd k).

In ManPG, the proximal parameter t controls the quadratic penalty in the tangent-space subproblem: A larger t weakens the penalty and encourages a larger step V_k. Adapting t via the BB principle requires differences of iterates and gradients. Working in the ambient Euclidean space $\mathbb{R}^{n\times r}$, we define

$$s_{k-1} = X_k - X_{k-1}, \qquad y_{k-1} = \nabla f(X_k) - \nabla f(X_{k-1}). \tag{7}$$

Ambient differences are used as in Riemannian spectral methods [31, 35]; the secant depends on the off-manifold extension of f (adding $\frac{a}{2}\|X\|_F^2$ is constant on $\mathcal{M}$ but shifts y), so t_k^{BB} is a curvature heuristic, not an intrinsic invariant. The GLL line search remains the safeguard.

Absolute-curvature policy. The signed guard $\langle s_{k-1}, y_{k-1}\rangle > 0$ freezes t on concave quadratics ($y_{k-1} = -2Ms_{k-1}$, $\langle s_{k-1}, y_{k-1}\rangle < 0$). We therefore use absolute curvature

$$t_k^{\mathrm{BB1}} = \frac{\|s_{k-1}\|_F^2}{|\langle s_{k-1}, y_{k-1}\rangle|}, \qquad t_k^{\mathrm{BB2}} = \frac{|\langle s_{k-1}, y_{k-1}\rangle|}{\|y_{k-1}\|_F^2}, \tag{8}$$

with the signed rule retained as an option (both coincide on CM). Equations (8)–(9) write the generic form at iteration k, where the pair (s_{k-1}, y_{k-1}) is defined in (7); Algorithm 1 applies the same rule to the fresh pair (s_k, y_k) in order to form t_{k+1}. Concretely, t_{k+1} is determined in each case as follows.

(a) Initialization. At the first iteration no secant information is available; we set $t_0 = 1/L$.

(b) BB1/BB2 alternation. From the pair (s_k, y_k) we form $t_{k+1}^{\mathrm{BB}} = \|s_k\|_F^2/|\langle s_k, y_k\rangle|$ (BB1) when k is even, and $t_{k+1}^{\mathrm{BB}} = |\langle s_k, y_k\rangle|/\|y_k\|_F^2$ (BB2) when k is odd. Alternation is a practical hedge: BB1 tends to overestimate and BB2 to underestimate the locally safe parameter; the sensitivity study in Section 5.3 comments on the choice.

(c) Numerically degenerate secants. When $|\langle s_k, y_k\rangle|$ vanishes (or the quotient underflows), we retain the previous value, $t_{k+1}^{\mathrm{BB}} = t_k$. With the absolute policy this occurs only when the secant is numerically zero; in the signed variant the same fallback applies whenever $\langle s_k, y_k\rangle \le 0$.

(d) Safeguarding. The quotient is projected onto a safe interval

$$t_k = \max(t_{\min},\ \min(t_{\max},\ t_k^{\mathrm{BB}})), \tag{9}$$

with $t_{\min} = 1/L$ and $t_{\max}$ a large positive constant (we use $t_{\max} = 10^4/L$); in Algorithm 1 the same projection produces t_{k+1} from t_{k+1}^{BB}.

The lower bound $t_{\min}$ keeps t_k from collapsing toward zero (which would force vanishing steps), while the upper bound $t_{\max}$ is what the descent analysis needs: Lemma 2 produces a uniform Armijo threshold for all $t_k \le t_{\max}$. Note that values $t_k > 1/L$ do *not* make the quadratic model a majorizer of f; descent is enforced by the line search rather than by majorization.

Remark 1. *ManPG-Ada [26] multiplies t by τ on accepted unit steps and divides otherwise (binary feedback versus a curvature estimate).*

3.2. Nonmonotone line search and the algorithm

BB stepsizes alone do not guarantee descent, particularly for nonconvex problems on curved domains. A monotone Armijo condition with BB steps often triggers excessive backtracking because the BB values naturally oscillate. The nonmonotone line search of Grippo, Lampariello, and Lucidi [30] (GLL), the scheme popularized for large-scale BB methods by Raydan [37], addresses this by comparing the trial value against the maximum over a sliding window of M preceding iterates.

With V_k obtained from (5) using $t = t_k$, we initialize $\alpha = 1$ and shrink $\alpha \leftarrow \gamma\alpha$ (with $\gamma \in (0, 1)$) until

$$F(\mathrm{Retr}_{X_k}(\alpha V_k)) \le \max_{0\le j\le \min(k,M-1)} F(X_{k-j}) \ - \ \delta\frac{\alpha}{2t_k}\|V_k\|_F^2, \tag{10}$$

where $\delta \in (0, 1)$ and $M \ge 1$ is the memory parameter. When $M = 1$, the condition reduces to a relaxed Armijo rule. In all experiments we use $M = 5$, $\gamma = 0.5$, and $\delta = 0.5$, which are common defaults in the nonmonotone literature [29, 30, 37].

GLL tolerates occasional objective increases so that the BB-prescribed t_k is accepted more often; $M = 1$ recovers monotone Armijo.

Algorithm 1 summarizes BB-ManPG. Lines 11–16 implement the absolute-curvature policy (8); the classical signed guard, in which t is updated only when $\langle s_k, y_k\rangle > 0$ and is left unchanged otherwise, is obtained by replacing the $|\langle s_k, y_k\rangle|$ test with a sign check, and is studied as a variant in Section 5.2. The subproblem in line 3 is solved exactly in the analysis; the inexact implementation of the inner solves is described in Remark 2. Computing s_k, y_k, and the BB quotient costs $O(nr)$, negligible next to the gradient evaluation and subproblem solve.

Algorithm 1 BB-ManPG for solving (1)

1: **Input:** $X_0 \in \mathrm{St}(n, r)$, $\gamma \in (0, 1)$, $\delta \in (0, 1)$, $M \in \mathbb{N}$ (memory), L (Lipschitz constant), $t_{\max} > 1/L$, $\epsilon > 0$
2: **Initialize:** $t_0 = 1/L$, $k = 0$
3: Compute $\nabla f(X_0)$, solve subproblem (5) with $t = t_0$ to obtain V_0
4: **while** $\|V_k\|_F / t_k > \epsilon$ **do**
5: $\quad \alpha \leftarrow 1, \quad \bar{F}_k \leftarrow \max\{F(X_{k-j}) : 0 \le j \le \min(k, M-1)\}$
6: $\quad$ **while** $F(\mathrm{Retr}_{X_k}(\alpha V_k)) > \bar{F}_k - \delta \frac{\alpha}{2t_k} \|V_k\|_F^2$ **do**
7: $\quad\quad \alpha \leftarrow \gamma\alpha$
8: $\quad$ **end while**
9: $\quad X_{k+1} \leftarrow \mathrm{Retr}_{X_k}(\alpha V_k)$
10: $\quad s_k \leftarrow X_{k+1} - X_k, \quad y_k \leftarrow \nabla f(X_{k+1}) - \nabla f(X_k)$
11: $\quad$ **if** $|\langle s_k, y_k\rangle| > 0$ **then**
12: $\quad\quad$ **if** k is even **then**
13: $\quad\quad\quad t_{k+1}^{\mathrm{BB}} \leftarrow \|s_k\|_F^2 \,/\, |\langle s_k, y_k\rangle|$
14: $\quad\quad$ **else**
15: $\quad\quad\quad t_{k+1}^{\mathrm{BB}} \leftarrow |\langle s_k, y_k\rangle| \,/\, \|y_k\|_F^2$
16: $\quad\quad$ **end if**
17: $\quad$ **else**
18: $\quad\quad t_{k+1}^{\mathrm{BB}} \leftarrow t_k$
19: $\quad$ **end if**
20: $\quad t_{k+1} \leftarrow \max(t_{\min}, \min(t_{\max}, t_{k+1}^{\mathrm{BB}}))$ $\quad$ where $t_{\min} = 1/L$
21: $\quad k \leftarrow k + 1$
22: $\quad$ Compute $\nabla f(X_k)$, solve subproblem (5) with $t = t_k$ to obtain V_k
23: **end while**

Remark 2 (Inexact subproblem solves)**.** *The analysis in Section 4 assumes exact solutions of* (5)*. In practice the dual subproblem is solved inexactly: For $r = 1$ by scalar bisection on the dual multiplier, and for $r > 1$ by the regularized semismooth Newton (SSN) method of Chen et al. [26, Section 4.2], terminated when the dual KKT residual $\|\mathrm{E}(\Lambda)\|_F^2 \le$* `inner_tol` *with an adaptive floor of order 10^{-9}–10^{-13}. A strictly positive accuracy floor does not by itself yield summable errors or an asymptotic stationarity theorem: The dual residual measures tangent-constraint feasibility, and a nonzero residual leaves $V = Z - X$ only approximately tangent. We therefore make no inexact-convergence claim here. All theorems below are exact-solve results; the finite tolerances are practical heuristics. For reference,*

an inexact convergence result along the lines of the inexact analyses in [17] would require (i) an explicit feasible primal residual (or a model-decrease condition) at each outer step, and (ii) a vanishing or relative forcing sequence on the dual residual, for instance $\|\mathrm{E}(\Lambda_k)\|_F \le \sigma \min\{\|V_k\|_F, 1\}$ *with* $\sigma \in (0,1)$ *independent of k, so that the per-step errors are controlled by the outer progress and become summable. In the experiments the inner tolerance is kept at an adaptive floor of order* 10^{-9}–10^{-13}*, i.e. several orders of magnitude below the outer stopping tolerance* $\epsilon = 10^{-6}nr$*, so that inner errors are negligible on the scale of the stopping test; within this regime the observed speedups are governed by the outer parameter and line-search choices rather than by the precise value of* `inner_tol`*. A full inexact theory is left for future work.*

4. Convergence analysis

This section establishes global convergence and iteration complexity for Algorithm 1. The proof structure parallels that of ManPG [26] but must handle the varying proximal parameter t_k and the nonmonotone acceptance criterion. The main technical task is to show that a uniform sufficient descent inequality holds for all $t_k \in [t_{\min}, t_{\max}]$, and that the nonmonotone condition still forces $\|V_k\|_F \to 0$.

Throughout this section we impose assumptions that match Chen et al. [26], with h required to be finite-valued.

Assumption 1. *The function* f *is continuously differentiable on* $\mathbb{R}^{n\times r}$ *and* ∇f *is L-Lipschitz continuous. The function h is finite-valued convex on* $\mathbb{R}^{n\times r}$ *(hence locally Lipschitz everywhere). Because* $\mathrm{St}(n,r)$ *is compact and h is locally Lipschitz, there exists* $H_0 > 0$ *such that* $\|\xi\|_F \le H_0$ *for all* $\xi \in \partial h(X)$ *and all* $X \in \mathcal{M}$*. Let* $G = \max_{X\in\mathcal{M}} \|\nabla f(X)\|_F < \infty$.

All theoretical guarantees below concern *exact* solutions of the tangent-space subproblem; finite inner tolerances are treated as implementation heuristics (see the remark after Algorithm 1).

4.1. Global convergence

We first collect auxiliary properties of the tangent-space subproblem and of the line search. Fix an iterate $X_k \in \mathcal{M}$ and proximal parameter $t_k > 0$. Define the subproblem objective

$$g_k(V) = \langle \nabla f(X_k), V\rangle + \frac{1}{2t_k}\|V\|_F^2 + h(X_k + V), \qquad V \in T_{X_k}\mathcal{M}. \tag{11}$$

The function g_k is $1/t_k$-strongly convex on $T_{X_k}\mathcal{M}$ because h is convex and the quadratic term provides the strong convexity modulus. The optimality condition for $V_k = \arg\min_{V\in T_{X_k}\mathcal{M}} g_k(V)$ reads

$$0 \in \frac{1}{t_k}V_k + \mathrm{grad} f(X_k) + \mathrm{Proj}_{T_{X_k}\mathcal{M}}(\partial h(X_k + V_k)), \tag{12}$$

where the inclusions are understood in $T_{X_k}\mathcal{M}$. Equivalently, there exists $w_k \in \partial h(X_k + V_k)$ with $0 = \frac{1}{t_k}V_k + \mathrm{grad} f(X_k) + \mathrm{Proj}_{T_{X_k}\mathcal{M}}(w_k)$, using $\mathrm{Proj}_{T_{X_k}\mathcal{M}}(\nabla f(X_k)) = \mathrm{grad} f(X_k)$ and $\langle \nabla f(X_k), V\rangle = \langle \mathrm{grad} f(X_k), V\rangle$ for $V \in T_{X_k}\mathcal{M}$.

Lemma 1. *For any* $\alpha \in [0,1]$*,*

$$g_k(\alpha V_k) - g_k(0) \le \frac{\alpha^2 - 2\alpha}{2t_k}\|V_k\|_F^2. \tag{13}$$

Proof. As in [26, Lemma 5.1] with t replaced by t_k: Strong convexity at V_k gives $g_k(0) \geq g_k(V_k) + \frac{1}{2t_k}\|V_k\|^2$, and convexity of h yields the bound for $\alpha \in [0, 1]$.

The next lemma extends [26, Lemma 5.2] to the case where t_k may vary across iterations. The key point is that a uniform lower bound on the Armijo stepsize α persists as long as t_k is bounded above.

Lemma 2. *Let $t_k \in [t_{\min}, t_{\max}]$ with $t_{\min} = 1/L$. There exists a constant $\bar{\alpha} > 0$, independent of k, such that for all $\alpha \in (0, \bar{\alpha}]$,*

$$F(\mathrm{Retr}_{X_k}(\alpha V_k)) - F(X_k) \leq -\frac{\alpha}{2t_k}\|V_k\|_F^2. \tag{14}$$

More precisely, $\bar{\alpha} = \min(1, 1/(2c_2 t_{\max}))$ with $c_2 = L/2 + (G_R + H_R)M_2$, where G_R and H_R bound $\|\nabla f\|$ and ∂h on a uniform ball containing all trial points (defined in the proof below; the same c_2 is used in Lemma 3 and Theorem 2).

Proof. Decompose the change in F into smooth and nonsmooth parts. For the smooth part, the Lipschitz continuity of ∇f gives

$$f(X_k + \alpha V_k) - f(X_k) \leq \alpha\langle \nabla f(X_k), V_k\rangle + \frac{L}{2}\alpha^2\|V_k\|_F^2. \tag{15}$$

First bound the tangent step without circularity. Exact model optimality and Lemma 1 (at $\alpha = 1$) give $\langle \mathrm{grad} f(X_k), V_k\rangle + h(X_k + V_k) - h(X_k) \leq -\|V_k\|_F^2/t_k$. Convexity yields $h(X_k + V_k) - h(X_k) \geq \langle z_k, V_k\rangle$ for any $z_k \in \partial h(X_k)$, so $\|V_k\|_F^2/t_k \leq -\langle \mathrm{grad} f(X_k) + z_k, V_k\rangle \leq (G + H_0)\|V_k\|_F$ and therefore

$$\|V_k\|_F \leq t_k(G + H_0) \leq V_{\max} := t_{\max}(G + H_0). \tag{16}$$

Every trial point $X_k + \alpha V_k$ ($\alpha \in [0, 1]$) and its retraction lie in the closed ball $\mathcal{B}_R$ of radius $R = \sqrt{r} + V_{\max}$ about the origin, which is convex and compact. Let $G_R = \max_{\mathcal{B}_R}\|\nabla f\|_F$ and H_R bound ∂h on $\mathcal{B}_R$. The retraction estimates (3) hold uniformly for $\|\xi\|_F \leq V_{\max}$ on the compact tangent disk (for the polar retraction, globally). Writing $\Delta := \|\mathrm{Retr}_{X_k}(\alpha V_k) - (X_k + \alpha V_k)\|_F \leq M_2\alpha^2\|V_k\|_F^2$, the segment joining the two points lies in $\mathcal{B}_R$, and

$$|f(\mathrm{Retr}_{X_k}(\alpha V_k)) - f(X_k + \alpha V_k)| \leq G_R \cdot M_2\alpha^2\|V_k\|_F^2. \tag{17}$$

Hence

$$f(\mathrm{Retr}_{X_k}(\alpha V_k)) - f(X_k) \leq \alpha\langle \nabla f(X_k), V_k\rangle + c_1\alpha^2\|V_k\|_F^2, \tag{18}$$

where $c_1 = L/2 + G_R M_2$.

For the nonsmooth term, the Lipschitz continuity of h on $\mathcal{B}_R$ together with (3) gives

$$h(\mathrm{Retr}_{X_k}(\alpha V_k)) - h(X_k) \leq h(X_k + \alpha V_k) - h(X_k) + H_R M_2\alpha^2\|V_k\|_F^2. \tag{19}$$

Adding (18) and (19),

$$F(\mathrm{Retr}_{X_k}(\alpha V_k)) - F(X_k) \leq \alpha\langle \nabla f(X_k), V_k\rangle + h(X_k + \alpha V_k) - h(X_k) + (c_1 + H_R M_2)\alpha^2\|V_k\|_F^2 \tag{20}$$

$$= g_k(\alpha V_k) - g_k(0) - \frac{\alpha^2}{2t_k}\|V_k\|_F^2 + (c_1 + H_R M_2)\alpha^2\|V_k\|_F^2 \tag{21}$$

$$\leq \Big(\frac{\alpha^2 - 2\alpha}{2t_k} - \frac{\alpha^2}{2t_k}\Big)\|V_k\|_F^2 + (c_1 + H_R M_2)\alpha^2\|V_k\|_F^2 \tag{22}$$

$$= -\frac{\alpha}{t_k}\|V_k\|_F^2 + (c_1 + H_R M_2)\alpha^2\|V_k\|_F^2 \tag{23}$$

$$= \Big((c_1 + H_R M_2)\alpha - \frac{1}{t_k}\Big)\alpha\|V_k\|_F^2, \tag{24}$$

where the second inequality uses Lemma 1.

Define $c_2 = c_1 + H_R M_2$ and set $\bar{\alpha} = \min(1,\ 1/(2c_2 t_{\max}))$. For any $\alpha \le \bar{\alpha}$ we have $\alpha \le 1/(2c_2 t_{\max}) \le 1/(2c_2 t_k)$ since $t_k \le t_{\max}$, so $c_2\alpha - 1/t_k \le -1/(2t_k)$, and (14) follows.

Lemma 2 shows that the monotone Armijo condition is satisfied for all sufficiently small α, with a uniform threshold $\bar{\alpha}$ that does not deteriorate as iterations proceed. Hence the backtracking loop in Algorithm 1 terminates after finitely many trials. Lemma 3 builds directly on this uniform threshold: Since every $\alpha \le \bar{\alpha}$ satisfies the GLL test, backtracking can stop only at or above $\bar{\alpha}$, and the accepted stepsizes inherit a uniform positive lower bound.

Lemma 3. *The accepted stepsize at iteration k satisfies $\alpha_k \ge \alpha_{\min} := \gamma\min(1, \bar{\alpha}) > 0$, with the same $\bar{\alpha} = \min(1, 1/(2c_2 t_{\max}))$ as in Lemma 2.*

Proof. If $\alpha = 1$ satisfies the GLL condition (10), then $\alpha_k = 1$. Otherwise, the backtracking loop shrinks α by γ repeatedly. For any $\alpha \le \bar{\alpha}$, Lemma 2 gives $F(\mathrm{Retr}_{X_k}(\alpha V_k)) - F(X_k) \le -\alpha/(2t_k)\|V_k\|_F^2$. Since $F(X_k) \le \bar{F}_k$ and $\delta \le 1$, this implies $F(\mathrm{Retr}_{X_k}(\alpha V_k)) \le \bar{F}_k - \delta\alpha/(2t_k)\|V_k\|_F^2$, i.e., the GLL condition holds for any $\alpha \le \bar{\alpha}$. Thus backtracking stops at or before the first $\alpha = \gamma^j$ with $\gamma^j \le \bar{\alpha}$. If $\bar{\alpha} < 1$, let j be the smallest integer with $\gamma^j \le \bar{\alpha}$. The loop stops at j (or earlier), giving $\alpha_k = \gamma^j > \gamma\bar{\alpha} = \alpha_{\min}$. If $\bar{\alpha} \ge 1$, the loop never backtracks and $\alpha_k = 1 \ge \gamma = \alpha_{\min}$.

We now prove that every limit point of $\{X_k\}$ is stationary. Define the window maximum

$$\bar{F}_k = \max_{0 \le j \le \min(k, M-1)} F(X_{k-j}), \qquad k \ge 0. \tag{25}$$

Lemma 4. *The sequence $\{\bar{F}_k\}$ is nonincreasing and bounded below, hence convergent to some $\bar{F}^*$.*

Proof. Boundedness follows from compactness of $\mathrm{St}(n, r)$ and continuity of F. For the nonincreasing property, observe that

$$\bar{F}_{k+1} = \max(F(X_{k+1}),\ \max_{1 \le j \le \min(k+1, M-1)} F(X_{k+1-j})). \tag{26}$$

The second maximum ranges over $F(X_k), \ldots, F(X_{k+1-\min(k+1, M-1)})$, all of which appear in the window defining $\bar{F}_k$. Hence $\bar{F}_{k+1} \le \max(F(X_{k+1}), \bar{F}_k)$. If the maximum is $F(X_{k+1})$, the GLL condition gives $F(X_{k+1}) \le \bar{F}_k - \delta\alpha_k/(2t_k)\|V_k\|_F^2 \le \bar{F}_k$; otherwise $\bar{F}_{k+1} \le \bar{F}_k$ directly.

Lemma 5. *For any $k \ge 0$, the window maximum satisfies the block-descent inequality*

$$\bar{F}_{k+M} \le \bar{F}_k - \min_{j \in \{k, \ldots, k+M-1\}} \delta\frac{\alpha_j}{2t_j}\|V_j\|_F^2. \tag{27}$$

Proof. Fix $j \in \{k, \ldots, k + M - 1\}$ and write $X_{j+1} = \mathrm{Retr}_{X_j}(\alpha_j V_j)$ for the point accepted at iteration j. Applying the GLL condition (10) at iteration j gives

$$F(X_{j+1}) \le \bar{F}_j - \delta\frac{\alpha_j}{2t_j}\|V_j\|_F^2, \tag{28}$$

and since $j \geq k$ and $\{\bar{F}_k\}$ is nonincreasing (Lemma 4), $\bar{F}_j \leq \bar{F}_k$, so that

$$F(X_{j+1}) \leq \bar{F}_k - \delta \frac{\alpha_j}{2t_j} \|V_j\|_F^2. \tag{29}$$

The window at iteration $k + M$ is exactly $\{k + 1, \ldots, k + M\}$ for every $k \geq 0$ (once $k + M \geq M - 1$ the window has length M and starts at $k + 1$; the finitely many early cases follow from the definition of $\bar{F}$). Every index in this window can be written as $j + 1$ for some $j \in \{k, \ldots, k + M - 1\}$. Therefore

$$\bar{F}_{k+M} = \max_{j \in \{k,\ldots,k+M-1\}} F(X_{j+1}) \tag{30}$$

$$\leq \max_{j \in \{k,\ldots,k+M-1\}} \Big(\bar{F}_k - \delta \frac{\alpha_j}{2t_j} \|V_j\|_F^2 \Big) \tag{31}$$

$$= \bar{F}_k - \min_{j \in \{k,\ldots,k+M-1\}} \delta \frac{\alpha_j}{2t_j} \|V_j\|_F^2, \tag{32}$$

which is exactly (27).

Theorem 1. *Consider the exact iteration of Algorithm 1 with the stopping test disabled (so the iteration runs indefinitely), with exact subproblem solutions and fixed window length $M \geq 1$. Then $\|V_k\|_F \to 0$, $F(X_k) \to \bar{F}^*$, and every accumulation point of $\{X_k\}$ is a stationary point of* (1).

Proof. By Lemmas 3–4, $\alpha_k \geq \alpha_{\min} > 0$, $t_k \leq t_{\max}$, and $\bar{F}_k \downarrow \bar{F}^*$. Put $c := \delta \alpha_{\min}/(2t_{\max}) > 0$. The GLL condition gives

$$F(X_{k+1}) \leq \bar{F}_k - c\|V_k\|_F^2. \tag{33}$$

For all large k choose $\ell(k) \in \{\max(0, k - M + 1), \ldots, k\}$ with $F(X_{\ell(k)}) = \bar{F}_k$. Since $\ell(k) \to \infty$ and, by (33) applied at $\ell(k) - 1$ when $\ell(k) \geq 1$,

$$0 \leq c\|V_{\ell(k)-1}\|_F^2 \leq \bar{F}_{\ell(k)-1} - F(X_{\ell(k)}) \to 0, \tag{34}$$

we obtain $V_{\ell(k)-1} \to 0$. The retraction bound then yields $\|X_{\ell(k)} - X_{\ell(k)-1}\|_F \to 0$, and uniform continuity of F on compact $\mathcal{M}$ gives $F(X_{\ell(k)-1}) \to \bar{F}^*$. Repeating the argument $q = 1, \ldots, M$ times shows $V_{\ell(k)-q} \to 0$ and $F(X_{\ell(k)-q}) \to \bar{F}^*$. Now fix a large index i and set $k = i + M$. Then $\ell(i + M) - i \in \{1, \ldots, M\}$, so $V_i = V_{\ell(i+M)-q}$ for some $q \in \{1, \ldots, M\}$ and $V_i \to 0$. Hence $\|V_k\|_F \to 0$ and $F(X_k) \to \bar{F}^*$.

Finally, let $X_{k_j} \to X_*$. Then $V_{k_j} \to 0$ and $X_{k_j} + V_{k_j} \to X_*$. The optimality condition (12) reads $0 = t_{k_j}^{-1} V_{k_j} + \operatorname{grad} f(X_{k_j}) + \operatorname{Proj}_{T_{X_{k_j}}\mathcal{M}}(w_{k_j})$ with $w_{k_j} \in \partial h(X_{k_j} + V_{k_j})$, $\|w_{k_j}\|_F \leq H_R$. Outer semicontinuity of the convex subdifferential and continuity of $\operatorname{grad} f$ and of the tangent projection along the subsequence give $0 \in \operatorname{grad} f(X_*) + \operatorname{Proj}_{T_{X_*}\mathcal{M}}(\partial h(X_*))$, i.e. (4).

Remark 3. *The proof uses only the uniform lower bound $\alpha_k \geq \alpha_{\min}$, the finite window, exact model minimization, and continuity of F on the compact manifold; it does not use the BB quotients themselves. It does not assert convergence of the whole sequence $\{X_k\}$ to a single point, and it does not cover the finite-tolerance implementation automatically (see the inexact-solve remark).*

4.2. Iteration complexity

Following [26], we say that X_k is ϵ-stationary if $\|V_k\|_F/t_k \le \epsilon$. When $V_k = 0$, the optimality condition (12) reduces to stationarity, so this is the natural measure. Since t_k varies across iterations and can be substantially larger than $1/L$ when the BB rule detects mild local curvature, the criterion $\|V_k\|_F/t_k \le \epsilon$ can be satisfied at a larger value of $\|V_k\|_F$ than the fixed-t criterion would allow. The stopping test is checked before each retraction step, and the iterate at which the test fires (the *tested iterate*) is the one returned by the implementation; this convention is used consistently in the experiments of Section 5. Throughout those experiments every run also logs the raw terminal $\|V_k\|_F$, and the terminal residual is *recomputed* by re-solving the subproblem at the returned iterate with a common reference parameter $t_{\mathrm{ref}} = 1/L$ for all methods, so that terminal accuracies are compared on one scale. The parameter t_{ref} serves only as a common reference for re-scoring and does not enter the iteration. Raw $\|V_k\|$ alone is not a parameter-independent certificate; the proximal residual at a common reference is the definition we use for cross-method comparisons (cf. the measure discussion in [26]).

Theorem 2. *Under Assumption 1 and exact subproblem solutions, Algorithm 1 with $t_{\min} = 1/L$ returns an ϵ-stationary point ($\|V_k\|_F/t_k \le \epsilon$) within*

$$K_\epsilon = \left\lceil \frac{M(F(X_0) - F^*)}{\delta\, C_{\mathrm{safe}}\, \epsilon^2} \right\rceil + M \tag{35}$$

outer iterations, where $F^* = \min_{X\in \mathrm{St}(n,r)} F(X)$ and $C_{\mathrm{safe}} = t_{\min}\alpha_{\min}/2$ with $\alpha_{\min} = \gamma \min(1, 1/(2c_2 t_{\max}))$ and $c_2 = L/2 + (G_R + H_R)M_2$ as in Lemma 2. The constant C_{safe} is the uniform lower bound of the quantity $\alpha_j t_j/2$: It converts the per-step decrease term $\delta\alpha_j\|V_j\|_F^2/(2t_j)$ into $\delta\, C_{\mathrm{safe}}(\|V_j\|_F/t_j)^2$, a squared residual scaled by an iteration-independent constant (see (36)*), which converts the telescoping decrease into the $O(\epsilon^{-2})$ count.*

Proof. By Lemma 3, $\alpha_j \ge \alpha_{\min}$ and $t_j \ge t_{\min}$, so $\alpha_j t_j/2 \ge C_{\mathrm{safe}}$ for every j, which is equivalent to

$$\delta\frac{\alpha_j}{2t_j}\|V_j\|_F^2 \ge \delta\, C_{\mathrm{safe}}\Big(\frac{\|V_j\|_F}{t_j}\Big)^2. \tag{36}$$

Applying the block bound of Lemma 5 with $k = iM$ and summing over the complete blocks $B_i = \{iM, \dots, (i+1)M - 1\}$ contained in $\{0, \dots, K-1\}$ gives

$$\sum_{i=0}^{N-1} \min_{j\in B_i} \delta\frac{\alpha_j}{2t_j}\|V_j\|_F^2 \le \bar{F}_0 - \bar{F}_{NM} \le F(X_0) - F^*, \tag{37}$$

where $N = \lfloor K/M \rfloor$ (the terms telescope because $\bar{F}_{(i+1)M}$ is the left-hand side of the bound for block i and the right-hand side for block $i+1$). If the algorithm has not terminated after K iterations, then $\|V_j\|_F/t_j > \epsilon$ for all $j = 0, \dots, K-1$, and (36) yields $\min_{j\in B_i} \delta\alpha_j\|V_j\|_F^2/(2t_j) \ge \delta C_{\mathrm{safe}}\epsilon^2$ for each complete block. Hence $N\delta C_{\mathrm{safe}}\epsilon^2 \le F(X_0) - F^*$. Taking $K = \lceil M(F(X_0) - F^*)/(\delta C_{\mathrm{safe}}\epsilon^2)\rceil + M$ gives $N > (F(X_0) - F^*)/(\delta C_{\mathrm{safe}}\epsilon^2)$, a contradiction. Therefore the stopping test must trigger within K_ϵ outer iterations.

The $O(\epsilon^{-2})$ rate matches ManPG [26, Theorem 5.5] and counts outer iterations with exact solves. The proof uses only $t_k \in [t_{\min}, t_{\max}]$, not the BB formulas, and applies to both curvature policies.

Remark 4. *When $M = 1$, the nonmonotone condition* (10) *reduces to $F(\mathrm{Retr}_{X_k}(\alpha V_k)) \le F(X_k) - \delta\alpha/(2t_k)\|V_k\|_F^2$, and Theorem 1 recovers the stationary-point conclusion of [26, Theorem 5.5] with a modified sufficient-decrease constant δ.*

5. Numerical experiments

All experiments use a single driver script (`run_unified_experiments.m`, MATLAB R2024a) with saved per-trial records (iterations, residuals at t_k and at $t_{\text{ref}} = 1/L$, convergence flags). Three confounds are controlled.

Matched starts. Every method and every grid point solves the same instance from the same X_0 after the same projected-subgradient warm start (500 steps for 1D CM and $r > 1$, 100 for 2D, 200 for power-law); warm-start cost is excluded from CPU time.

Aligned acceptance tests. All solvers use the same decrease term $\delta\alpha\|V_k\|_F^2/(2t)$ with $\delta = \gamma = 0.5$. ManPG/Ada are monotone; BB-ManPG and nonmonotone baselines use GLL with $M = 5$. Coefficient alignment is treated as a design requirement.

Comparable stopping. Tolerance $\epsilon = 10^{-6}nr$ on $\|V_k\|_F/t_k$ is checked before each retraction step and the tested iterate is returned; residuals are also logged at $t_{\text{ref}} = 1/L$. Budgets: 2000 (1D CM, power-law) or 3000 (2D, $r > 1$). Ten trials; capped runs are counted at the budget and reported explicitly. Statistics are mean ± sd.

Statistics. Outer-iteration counts are compared across the ten matched trials. A paired win for BB-ManPG against a competitor means that BB-ManPG needs strictly fewer outer iterations on the same starting point (exact ties count neither as wins nor as losses). The reported p-values are two-sided Welch t-tests on the two samples of per-trial iteration counts (unequal variances, Welch–Satterthwaite degrees of freedom); capped runs enter the test at the budget value. All win counts and p-values cited in the text are tabulated in the notes to Tables 1–4.

Methods. BB-ManPG: Absolute-curvature alternating BB on t_k, $M = 5$, $\gamma = \delta = 0.5$, $t_{\max} = 10^4/L$ (signed variant in Section 3.1). ManPG: $t = 1/L$. ManPG-Ada: $\tau = 1.01$. NPGL-type: Fixed $t = 1/L$, same-interval secants $S = X_k - X_{k-1}$, $Y = d_k - d_{k-1}$, absolute BB on the *outer* step with $\alpha \in [10^{-10}, 10^{10}]$, GLL. PQN-s: Scalar $\mathcal{B}_k = t_k^{-1}I$ with BB2 scale, GLL. Both are our reimplementations, not the authors' full codes.

Inner solvers. Bisection for $r = 1$; SSN [26, Section 4.2] for $r > 1$; thin-SVD polar retraction. Inner counts are not comparable across r.

5.1. Compressed modes

The CM problem on the sphere is

$$\min_{\|x\|_2=1} \; x^\top Hx + \mu\|x\|_1, \tag{38}$$

where H is the discrete 1D free-electron Hamiltonian on $[0, 50]$ with n grid points [1], with $\mu = 0.10$ and $L = 2\lambda_{\max}(H)$. Because the largest eigenvalue of the discretized Laplacian scales as $O(n^2)$, the default $t = 1/L$ becomes vanishingly small as n grows, which makes this class the natural stress test for parameter adaptation. On this class the smooth part is a convex quadratic, so $\langle s_k, y_k\rangle = 2s_k^\top Hs_k \geq 0$ along every step and the absolute-curvature rule coincides with the classical signed rule; the BB and BBsgn iteration vectors are identical in all 40 runs of this block, and BBsgn is omitted from Table 1.

Table 1 reports the results. BB-ManPG needs the fewest iterations at every size, with paired wins of 10/10 against ManPG at all four sizes (two-sided Welch $p = 0.016$, 2.9×10^{-5}, 1.1×10^{-12}, 1.6×10^{-5} at $n = 32, 64, 128, 256$): Reductions of about 2.9×, 5.6×, 5.2×, and 8.6× at $n = 32, 64, 128, 256$. At $n = 128$ ManPG and NPGL-type fail to meet tolerance in all ten trials (0/10), while BB-ManPG

converges in every one. ManPG-Ada improves on ManPG but remains 1.5–5.1× slower than BB-ManPG (paired BB wins 10/10 at every size). PQN-s is the strongest competitor and stays within 1.1–1.5× of BB-ManPG (paired BB wins 10, 10, 10, 9 of 10). Inner-solver counts are nearly identical across methods (bisection, 32–40 per outer step), so the outer-iteration ratios carry over to work. Terminal residuals at the common reference $t_{\text{ref}} = 1/L$ are $(2.6, 6.3, 13.4, 25.9) \times 10^{-5}$ for BB-ManPG against tolerances $(3.2, 6.4, 12.8, 25.6) \times 10^{-5}$: Passing its own stopping test does not certify the common reference threshold, and terminal residuals must be judged at a common reference rather than on each method's own adaptive scale. Final objectives across methods agree with the per-trial best to within 2×10^{-9} (n=32) up to 2×10^{-7} (n=256).

Table 1. 1D CM from the unified records ($r = 1$, $\mu = 0.10$, 10 trials, 2000-iteration budget, matched starts). Outer iterations, mean ± sd (converged/10); capped runs counted at the budget.

n	L	BB-ManPG	ManPG	Ada	NPGL-type	PQN-s
32	1.7	71 ± 48 (10)	204 ± 140 (10)	108 ± 48 (10)	352 ± 509 (10)	76 ± 53 (10)
64	6.8	243 ± 80 (10)	1368 ± 472 (10)	609 ± 194 (10)	1475 ± 474 (9)	327 ± 109 (10)
128	27	381 ± 93 (10)	2000 ± 0 (0)	1387 ± 316 (10)	2000 ± 0 (0)	518 ± 122 (10)
256	106	45 ± 10 (10)	386 ± 130 (10)	231 ± 69 (10)	629 ± 512 (9)	65 ± 19 (10)

*Note: BBsgn is omitted from this table: On the convex-quadratic CM class the signed and absolute-curvature rules coincide, so BBsgn reproduces the BB-ManPG iteration vector in all 40 runs of this block. Paired wins of BB-ManPG (out of 10) and two-sided Welch p over the ten matched trials: Vs. ManPG 10/10 at every size ($p = 0.016, 2.9 \times 10^{-5}, 1.1 \times 10^{-12}, 1.6 \times 10^{-5}$ for $n = 32, 64, 128, 256$); vs. ManPG-Ada 10/10 at every size ($p = 0.10, 1.4 \times 10^{-4}, 1.4 \times 10^{-6}, 1.1 \times 10^{-5}$); vs. PQN-s 10, 10, 10, 9 of 10 ($p = 0.81, 0.067, 0.012, 0.009$).

To check that the advantage scales with dimension, we extend CM to the 2D domain $[0, 1]^2$ with periodic boundary conditions. The 2D free-electron Hamiltonian is $H_{2D} = I_{n_y} \otimes H_{1D,x} + H_{1D,y} \otimes I_{n_x}$ with $n_x = n_y$ and total dimension $n = n_x^2$; the unit domain gives $L = 2\lambda_{\max}(H_{2D}) = 8n_x^2$, ranging from 2.0×10^3 to 1.8×10^4, so the default parameter $t = 1/L \in [5.4 \times 10^{-5}, 4.9 \times 10^{-4}]$ is pathologically small. We use $\mu = 0.10$, $r = 1$, a 3000-iteration budget, and 100 warm-start iterations.

Table 2. 2D CM on $[0, 1]^2$ ($r = 1$, $\mu = 0.10$, 10 trials, 3000-iteration budget). Outer iterations, mean ± sd (converged/10).

n_x	n	L	BB-ManPG	ManPG	ManPG-Ada	PQN-s
16	256	2.0×10^3	49 ± 6 (10)	504 ± 49 (10)	245 ± 20 (10)	57 ± 7 (10)
24	576	4.6×10^3	83 ± 10 (10)	1256 ± 54 (10)	582 ± 25 (10)	99 ± 14 (10)
32	1024	8.2×10^3	128 ± 14 (10)	2269 ± 224 (10)	1046 ± 105 (10)	161 ± 29 (10)
48	2304	1.8×10^4	239 ± 50 (10)	3000 ± 0 (0)	2518 ± 230 (10)	397 ± 99 (10)

*Note: NPGL-type is omitted from this table; its iteration counts match ManPG up to rounding at every n_x (503.5, 1256.1, 2269.4, 3000 ± 0 (0)). Paired wins of BB-ManPG (out of 10) and two-sided Welch p: Vs. ManPG 10/10 at every size ($p = 2.1 \times 10^{-10}, 3.8 \times 10^{-14}, 2.1 \times 10^{-10}, 3.4 \times 10^{-17}$ for $n_x = 16, 24, 32, 48$); vs. ManPG-Ada 10/10 at every size ($p = 1.3 \times 10^{-11}, 8.6 \times 10^{-16}, 3.1 \times 10^{-10}, 4.3 \times 10^{-11}$); vs. PQN-s 7, 8, 9, 10 of 10 ($p = 0.0056, 0.0099, 0.0064, 5.6 \times 10^{-4}$).

Table 2 reports the results. BB-ManPG reduces outer iterations by about 10.4×, 15.2×, and 17.8× against ManPG at $n_x = 16, 24, 32$ (paired 10/10 wins; Welch $p = 2.1 \times 10^{-10}, 3.8 \times 10^{-14}, 2.1 \times 10^{-10}$, respectively), and at $n_x = 48$ ManPG and NPGL-type exhaust the budget in all ten trials while BB-ManPG converges in every trial in 239 ± 50 iterations. ManPG-Ada is 5–10.5× slower than BB-ManPG; PQN-s is again second at

1.2–1.7× BB. Bisection counts are again nearly identical across methods (28.0–32.1 per outer step). CPU means track the iteration counts at the larger sizes (at $n_x = 48$: BB 0.41 s vs. ManPG 3.22 s and ManPG-Ada 3.30 s); final objectives agree with the per-trial best to within 2×10^{-8} among converged runs. The signed variant again coincides with BB-ManPG on this convex-quadratic class.

Table 3. CM with $r > 1$ ($\mu = 0.10$, 10 trials, 3000-iteration budget). Outer iterations, mean ± sd (converged/10).

n	r	BB-ManPG	ManPG	ManPG-Ada	PQN-s
64	2	637 ± 848 (9)	2111 ± 873 (6)	1603 ± 1214 (6)	540 ± 260 (10)
128	2	1123 ± 1298 (7)	2570 ± 612 (6)	1723 ± 922 (7)	570 ± 264 (10)
64	4	1193 ± 1248 (7)	3000 ± 0 (0)	2775 ± 711 (1)	635 ± 251 (10)
128	4	2468 ± 1122 (2)	2886 ± 360 (1)	2792 ± 659 (1)	1131 ± 745 (9)

*Note: Paired wins of BB-ManPG (out of 10; exact ties at the budget are not counted) and two-sided Welch p, in the order of the rows: Vs. ManPG 9, 7, 7, 2 (p = 0.0012, 0.0072, 0.0013, 0.29); vs. ManPG-Ada 9, 7, 7, 2 (p = 0.056, 0.25, 0.0036, 0.44); vs. PQN-s 9, 7, 7, 2 (p = 0.74, 0.22, 0.20, 0.0065, with PQN-s faster at $(128, 4)$).

Table 3 repeats CM with $r = 2, 4$ ($n = 64, 128$; SSN inner solver). BB-ManPG beats ManPG and ManPG-Ada on every configuration, but conservative PQN-s is most robust at higher rank (9–10/10) and 1.2–2.2× faster than the alternating rule; at $(128, 4)$ only PQN-s converges reliably. The increasingly aggressive BB1 values also trigger backtracking more often as r grows, which explains part of the convergence failures above. The SSN cost per outer step grows with r, so CPU time no longer tracks outer iterations exactly (e.g., 2.33 s vs. 0.32 s for BB-ManPG and PQN-s at $(128, 4)$, where the latter uses fewer but similarly priced steps); final-objective gaps to the per-trial best are at most 7×10^{-4} for BB-ManPG and 7×10^{-3} across all methods on these blocks.

The higher-rank behavior calls for a conservative safeguard rather than for abandoning the adaptive parameter, and three practical mitigations are available. First, one can restrict the update to BB2 (the short, conservative step) when $r > 1$: The BB-strategy records (Section 5.3) identify BB2 as the conservative choice, trading somewhat more outer iterations for far fewer backtracks. Second, one can lower $t_{\max}$ as r grows, so that the bold BB1 values are clipped before they trigger long backtracking cascades. Third, one can reset $t_k = 1/L$ after a fixed number of consecutive rejected unit steps, which restores the ManPG regime exactly where the adaptive rule becomes unreliable; a hybrid that falls back to the conservative PQN-s scale after repeated failures is a further possibility. A rank-adaptive safeguard policy is left for future work.

5.2. Power-law spectrum problem

To probe whether the acceleration extends beyond the Schrödinger structure, we study ℓ_1-regularized PCA on the sphere,

$$\min_{x \in \mathbb{S}^{n-1}} \; -x^\top M x + \mu \|x\|_1, \tag{39}$$

where $M = Q\,\mathrm{diag}(1^2, 2^2, \ldots, n^2)Q^\top$ with a random dense orthogonal Q and $L = 2n^2$. This class is the decisive test for the curvature policy of Section 3.1: Since $y_k = -2Ms_k$ and $\langle s_k, y_k \rangle = -2s_k^\top M s_k < 0$ whenever $s_k \neq 0$, the classical signed guard never updates t_k, and every method equipped with it freezes at $t = 1/L$. Table 4 makes the diagnosis quantitative: The signed variant records exactly 0 BB updates

in every trial of this block, and its iteration vector is *identical* to those of NPGL-type and PQN-s (whose signed guards also never fire), so the three rows coincide. The absolute-curvature rule (8) updates t_k at essentially every iteration (BB-update counts 125–1568 per run).

Table 4. Power-law PCA from the unified records ($r = 1$, 10 trials, 2000-iteration budget). Outer iterations, mean ± sd (converged/10). BBsgn, NPGL-type and PQN-s coincide on this class.

n	μ	BB-ManPG	BBsgn	ManPG	Ada
64	0.05	177 ± 21 (10)	205 ± 18 (10)	412 ± 561 (9)	117 ± 11 (10)
64	0.10	175 ± 26 (10)	220 ± 20 (10)	588 ± 745 (8)	121 ± 9 (10)
128	0.05	182 ± 24 (10)	569 ± 58 (10)	1094 ± 628 (7)	471 ± 538 (9)
128	0.10	208 ± 74 (10)	595 ± 110 (10)	1500 ± 656 (4)	313 ± 49 (10)
256	0.05	559 ± 532 (10)	1586 ± 183 (9)	2000 ± 0 (0)	1794 ± 435 (2)
256	0.10	522 ± 379 (10)	1489 ± 100 (10)	2000 ± 0 (0)	1970 ± 96 (1)

*Note: Complete trial-level arrays are in `unified_D.mat`. Paired wins of BB-ManPG (out of 10) and two-sided Welch p, in the order of the rows: Vs. BBsgn 8, 10, 10, 10, 10, 10 ($p = 0.0042$, 5.2×10^{-4}, 1.7×10^{-10}, 9.4×10^{-8}, 1.2×10^{-4}, 1.3×10^{-5}); vs. ManPG-Ada 0, 0, 10, 9, 9, 10 (ManPG-Ada is faster at $n = 64$; $p = 1.3 \times 10^{-6}$, 6.8×10^{-5}, 0.12, 0.0018, 2.5×10^{-5}, 3.2×10^{-7}).

Table 5. Terminal proximal residuals at the common reference $t_{\text{ref}} = 1/L$ for the power-law block of Table 4 (mean over the ten trials; nominal tolerance $\epsilon = 10^{-6}n$).

n	μ	BB-ManPG	BBsgn†	ManPG	Ada
64	0.05	6.5×10^{-4}	6.1×10^{-5}	6.4×10^{-5}	5.7×10^{-5}
64	0.10	4.7×10^{-4}	6.1×10^{-5}	6.4×10^{-5}	5.9×10^{-5}
128	0.05	1.2×10^{-3}	1.3×10^{-4}	1.5×10^{-4}	1.3×10^{-4}
128	0.10	1.1×10^{-3}	1.3×10^{-4}	1.6×10^{-4}	1.2×10^{-4}
256	0.05	3.5×10^{-3}	2.6×10^{-4}	9.7×10^{-4}	3.5×10^{-4}
256	0.10	3.4×10^{-3}	2.5×10^{-4}	1.1×10^{-3}	4.0×10^{-4}

*Note: †NPGL-type and PQN-s share the BBsgn column on this class (identical iteration vectors). Capped runs (ManPG at $n = 256$) are scored at the budget iterate. BBsgn's re-scored residuals stay within the nominal tolerance in every row, whereas BB-ManPG's sit 7–14× above it despite meeting their own stopping tests; ManPG-Ada's exceed the tolerance in three of the four configurations with $n \geq 128$.

Because starts, tolerance, and decrease coefficients are matched, the effects of the acceptance rule and of the parameter update can be separated. With nonmonotone acceptance alone, BB-signed (≡ fixed $1/L$ + GLL) improves on monotone ManPG at $n \geq 128$ (568.6 vs. 1093.8 at $\mu = 0.05$) but still runs close to the budget at $n = 256$ (1489–1587 iterations, with one failure at $\mu = 0.05$). With active parameter adaptation, BB-ManPG (abs) is 1.2–3.1× faster than its signed twin (paired wins 8–10/10; Welch $p \leq 0.0043$ in every block, down to 1.7×10^{-10}) and is the only method converging in 10/10 trials in both μ-configurations at $n = 256$, where ManPG fails in all trials, ManPG-Ada fails in 8–9 of 10, and no single fixed c reaches 10/10 at $\mu = 0.05$ (the best, $c = 20$, reaches 7/10). Several qualifications are in order. ManPG-Ada is the fastest method at $n = 64$ (117–121 iterations vs. 175–177 for BB-ManPG, paired 0/10), and the per-trial oracle best fixed t is faster still at $n \leq 128$ (79–211 iterations), although which c is best varies with n and μ and the oracle is unavailable in practice. Backtracking is frequent under the bold parameters ($\alpha = 1$ accepted in only 14–52% of BB-ManPG steps on this class, vs. 96–100% for the frozen variants). BB-ManPG's terminal residuals at the common reference $t_{\text{ref}} = 1/L$

are 7–14× larger than the tolerance on this class (4.7×10^{-4} to 3.5×10^{-3}) (Table 5), even though all runs met their own stopping test and the final objectives agree with the per-trial best to within 1.5×10^{-9}; the residual is scored at the tested iterate returned by the implementation (Section 4.2).

5.3. Ablations

To separate curvature adaptation from a merely enlarged fixed parameter, every trial of the 1D CM block also runs ManPG with fixed $t = c/L$ for $c \in \{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ from the *same* initial points. On the same starts, the best fixed constant at each size ($c = 20$ for $n = 32, 64$ and $c = 10$ for $n = 128, 256$) gives means 49, 257, 833, 122 versus BB-ManPG 71, 243, 381, 45 (full grid in `unified_A.mat`). At $n = 32$ a well-chosen constant even beats BB. The advantage of the BB rule is that the best constant varies with n, μ, and the problem class, whereas the BB rule finds a comparably good scale automatically, without per-problem tuning.

Table 6. Four-way ablation on 1D CM (10 matched starts, 2000 budget, aligned decrease coefficients). Outer iterations, mean ± sd (converged/10); α_1 fraction of unit steps.

Stepsize	Line search	$n = 64$ Iter	α_1	$n = 256$ Iter	α_1
Fixed $10/L$	Monotone	337 ± 35 (10)	0.35	124 ± 87 (10)	0.16
Fixed $10/L$	GLL	494 ± 49 (10)	0.12	383 ± 569 (9)	0.02
BB-ManPG	Monotone	304 ± 32 (10)	0.84	44 ± 12 (10)	0.83
BB-ManPG	GLL	285 ± 32 (10)	0.99	44 ± 13 (10)	0.91

To separate the parameter update from the acceptance rule, four configurations were run on 1D CM with matched starts and aligned coefficients (10 matched starts): Fixed $t = 10/L$ with monotone Armijo or GLL, and BB (abs) with monotone Armijo or GLL. Table 6 reports the outcome. The parameter update dominates (fixed → BB cuts iterations at both sizes). GLL helps BB slightly at $n = 64$ and is neutral at $n = 256$, but hurts a fixed over-large t. BB+GLL accepts unit steps in 91–99% of iterations.

The method is insensitive to $M \in \{2, \ldots, 50\}$ (mean iterations 249–265 on 1D CM at $n = 64$), $\gamma, \delta \in [0.1, 0.9]$ (248–251), and $t_{\max} \in \{10^2, 10^4, 10^6\}/L$ (identical counts). Cold-start and BB1/BB2/alternating comparisons in the released records show the same qualitative picture: The BB parameter update dominates; BB1 is fastest with more backtracking, BB2 is conservative, and alternation is a safe default; the warm start contributes most at $n = 256$ (from 128 to 46 iterations) and is essentially neutral at $n = 128$.

On compressed modes, where L grows quadratically and $\langle s, y \rangle > 0$ along the path, BB-ManPG cuts iterations by roughly 3–9× against ManPG in 1D and 10–18× in 2D. The fixed-t grid shows that most of the gap is closed by any sensibly enlarged constant; the BB rule supplies that scale automatically and is strongest where $t = 1/L$ is badly wrong. ManPG-Ada eventually escapes $1/L$ but pays for the climb. At higher rank the conservative PQN-s is more robust than the alternating rule, a real trade-off.

On the power-law PCA class the signed guard never updates t (BBsgn, NPGL-type and PQN-s coincide), while the absolute-curvature rule is the only one that adapts and is the most reliable at $n = 256$ (10/10 own-test successes vs. 0/10 for monotone ManPG). The costs are frequent backtracking under bold t_k and terminal residuals that must be judged at a common reference.

With aligned decrease constants the parameter update dominates (Table 6); GLL is a modest safeguard for BB and can hurt a fixed over-large t. We recommend the simpler BB + monotone Armijo variant

when the adaptive t_k is already well safeguarded and unit steps are accepted at a high rate, as on the compressed-modes blocks, where BB+GLL accepts $\alpha = 1$ in 91–99% of iterations and the monotone variant stays within 7% of it. GLL is preferable when the BB values oscillate strongly, as on the power-law class, where bold t_k values make backtracking frequent and a monotone test would reject the temporary increases that the nonmonotone window absorbs. We keep BB+GLL as the default because it is the safer choice across problem classes, and treat BB+monotone as a viable simpler variant.

6. Conclusion

We adapt the Barzilai–Borwein rule to the proximal parameter of ManPG using absolute ambient secant curvature, pair it with a GLL line search, and prove global convergence to stationary points with $O(\epsilon^{-2})$ complexity for exact subproblem solves and any safeguarded $t_k \in [1/L, t_{\max}]$. Under a matched protocol, BB-ManPG is consistently faster than ManPG and ManPG-Ada on compressed modes and is the only adaptive method among those tested that converges in all trials on the concave power-law PCA class, where signed guards freeze. Ablations show that the parameter update, not the nonmonotone window, is the primary accelerator. The approach has two limitations. First, t_k^{BB} is computed from ambient secants and therefore depends on the off-manifold extension of f (adding $\frac{a}{2}\|X\|_F^2$ is constant on $\mathcal{M}$ but shifts y), so it is a curvature heuristic rather than an intrinsic invariant. Second, all theoretical guarantees are for exact subproblem solves; extending the convergence and complexity analysis to the finite-tolerance implementation, for instance under a relative forcing sequence on the dual residual as discussed in Remark 2, is left for future work.

Author contributions

Conceptualization, D. Tang and L. Chen; methodology, D. Tang and L. Chen; software, D. Tang; validation, D. Tang and L. Chen; formal analysis, D. Tang and L. Chen; investigation, D. Tang and L. Chen; resources, L. Chen; data curation, D. Tang; writing—original draft preparation, D. Tang; writing—review and editing, D. Tang and L. Chen; visualization, D. Tang; supervision, L. Chen; project administration, L. Chen; funding acquisition, L. Chen.

Use of Generative-AI tools declaration

Generative AI tools were used in the preparation of this manuscript solely for polishing the writing and checking typographical errors. The authors have reviewed and take full responsibility for the final content.

Acknowledgments

This work was partially supported by the Science and Technology Research Program of Chongqing Municipal Education Commission (KJQN202000533, KJZD-M202300504), the Chongqing Natural Science Foundation Project (CSTB2024NSCQ-LZX0140, CSTB2025NSCQ-LZX0008), and the National Natural Science Foundation of China (12171063). The MATLAB source code and archived per-run experimental records supporting the numerical results are available from the authors on reasonable request.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. V. Ozoliňš, R. Lai, R. Caflisch, S. Osher, Compressed modes for variational problems in mathematics and physics, *Proc. Natl. Acad. Sci. USA,* **110** (2013), 18368–18373. https://doi.org/10.1073/pnas.1318679110
2. H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, *J. Comput. Graph. Statist.,* **15** (2006), 265–286. https://doi.org/10.1198/106186006X113430
3. Y. Yang, H. T. Shen, Z. Ma, Z. Huang, X. Zhou, $\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning, *Proc. 22nd Int. Joint Conf. Artif. Intell.,* (2011), 1589–1593. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-267
4. Y. Zhang, Y. Lau, H. W. Kuo, S. Cheung, A. Pasupathy, J. Wright, On the global geometry of sphere-constrained sparse blind deconvolution, *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* (2017), 4381–4389. https://doi.org/10.1109/CVPR.2017.466
5. P. A. Absil, R. Mahony, R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton: Princeton University Press, 2009. https://doi.org/10.1515/9781400830244
6. N. Boumal, P. A. Absil, C. Cartis, Global rates of convergence for nonconvex optimization on manifolds, *IMA J. Numer. Anal.,* **39** (2019), 1–33. https://doi.org/10.1093/imanum/drx080
7. J. Hu, X. Liu, Z. Wen, Y. Yuan, A brief introduction to manifold optimization, *J. Oper. Res. Soc. China,* **8** (2020), 199–248. https://doi.org/10.1007/s40305-020-00295-9
8. O. P. Ferreira, P. R. Oliveira, Subgradient algorithm on Riemannian manifolds, *J. Optim. Theory Appl.,* **97** (1998), 93–104. https://doi.org/10.1023/A:1022675100677
9. S. Hosseini, A. Uschmajew, A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds, *SIAM J. Optim.,* **27** (2017), 173–189. https://doi.org/10.1137/16M1069298
10. X. Li, Y. Zhong, The stochastic projected gradient method for optimization problems on Hadamard manifolds, *Optim. Erudit.,* **3** (2026), 20–33. https://doi.org/10.69829/oper-026-0301-ta03
11. O. P. Ferreira, P. R. Oliveira, Proximal point algorithm on Riemannian manifolds, *Optimization,* **51** (2002), 257–270. https://doi.org/10.1080/02331930290019413
12. G. C. Bento, J. X. Cruz Neto, P. R. Oliveira, A new approach to the proximal point method: Convergence on general Riemannian manifolds, *J. Optim. Theory Appl.,* **168** (2016), 743–755. https://doi.org/10.1007/s10957-015-0861-2
13. G. C. Bento, C. R. Santiago, On the convergence and properties of a proximal-gradient method on Hadamard manifolds, *J. Nonlinear Var. Anal.,* **10** (2026), 617–633. https://doi.org/10.23952/jnva.10.2026.3.06
14. R. Lai, S. Osher, A splitting method for orthogonality constrained problems, *J. Sci. Comput.,* **58** (2014), 431–449. https://doi.org/10.1007/s10915-013-9740-x

15. W. Chen, H. Ji, Y. You, An augmented Lagrangian method for ℓ_1-regularized optimization problems with orthogonality constraints, *SIAM J. Sci. Comput.,* **38** (2016), B570–B592. https://doi.org/10.1137/140988875

16. A. Kovnatsky, K. Glashoff, M. M. Bronstein, MADMM: A generic algorithm for non-smooth optimization on manifolds, *Proc. European Conf. Comput. Vis.,* (2016), 680–696. https://doi.org/10.1007/978-3-319-46454-1_41

17. K. Deng, Z. Peng, A manifold inexact augmented Lagrangian method for nonsmooth optimization on Riemannian submanifolds in Euclidean space, *IMA J. Numer. Anal.,* **43** (2023), 1653–1684. https://doi.org/10.1093/imanum/drac018

18. W. Huang, K. Wei, Riemannian proximal gradient methods, *Math. Program.,* **194** (2022), 371–413. https://doi.org/10.1007/s10107-021-01632-3

19. W. Huang, K. Wei, An extension of fast iterative shrinkage-thresholding algorithm to Riemannian optimization for sparse principal component analysis, *Numer. Linear Algebra Appl.,* **29** (2022), e2409. https://doi.org/10.1002/nla.2409

20. B. Gao, X. Liu, X. Chen, Y. Yuan, A new first-order algorithmic framework for optimization problems with orthogonality constraints, *SIAM J. Optim.,* **28** (2018), 302–332. https://doi.org/10.1137/16M1098759

21. N. Xiao, X. Liu, Y. Yuan, Exact penalty function for $\ell_{2,1}$ norm minimization over the Stiefel manifold, *SIAM J. Optim.,* **31** (2021), 3097–3126. https://doi.org/10.1137/20M1354313

22. N. Xiao, X. Liu, K. C. Toh, Dissolving constraints for Riemannian optimization, *Math. Oper. Res.,* **49** (2023), 366–397. https://doi.org/10.1287/moor.2023.1360

23. X. Hu, N. Xiao, X. Liu, K. C. Toh, A constraint dissolving approach for nonsmooth optimization over the Stiefel manifold, *IMA J. Numer. Anal.,* **44** (2024), 3717–3748. https://doi.org/10.1093/imanum/drad098

24. X. Liu, N. Xiao, Y. Yuan, A penalty-free infeasible approach for a class of nonsmooth optimization problems over the Stiefel manifold, *J. Sci. Comput.,* **99** (2024), 30. https://doi.org/10.1007/s10915-024-02495-4

25. N. Xiao, X. Liu, Solving optimization problems over the Stiefel manifold by smooth exact penalty function, *J. Comput. Math.,* **42** (2024), 1246–1276. https://doi.org/10.4208/jcm.2307-m2021-0331

26. S. Chen, S. Ma, A. M. C. So, T. Zhang, Proximal gradient method for nonsmooth optimization over the Stiefel manifold, *SIAM J. Optim.,* **30** (2020), 210–239. https://doi.org/10.1137/18M122457X

27. Y. Wang, W. Yin, J. Zeng, Global convergence of ADMM in nonconvex nonsmooth optimization, *J. Sci. Comput.,* **78** (2019), 29–63. https://doi.org/10.1007/s10915-018-0757-z

28. 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

29. S. J. Wright, R. D. Nowak, M. A. T. Figueiredo, Sparse reconstruction by separable approximation, *IEEE Trans. Signal Process.,* **57** (2009), 2479–2493. https://doi.org/10.1109/TSP.2009.2016892

30. L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM J. Numer. Anal.,* **23** (1986), 707–716. https://doi.org/10.1137/0723046

31. 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

32. Z. Ma, T. Yan, S. Zhao, Global convergence of a Barzilai–Borwein scaling HZ-type Riemannian conjugate gradient method, *J. Appl. Math. Inform.,* **44** (2026), 921–943.

33. Q. Wang, W. H. Yang, Proximal quasi-Newton method for composite optimization over the Stiefel manifold, *J. Sci. Comput.,* **95** (2023), 39. https://doi.org/10.1007/s10915-023-02165-x

34. J. C. Zhang, J. Gao, Y. K. Huang, X. W. Liu, Two nonmonotone proximal gradient methods for nonsmooth optimization over the Stiefel manifold, *Acta Math. Appl. Sin. Engl. Ser.,* **42** (2026), 105–120. https://doi.org/10.1007/s10255-024-1065-z

35. 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

36. R. Fletcher, On the Barzilai–Borwein method, *Optim. Control Appl.,* (2005), 235–256. https://doi.org/10.1007/0-387-24255-4_10

37. M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM J. Optim.,* **7** (1997), 26–33. https://doi.org/10.1137/S1052623494266365



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)