Optim Lett (2016) 10:1705–1723
DOI 10.1007/s11590-015-0977-1

ORIGINAL PAPER

A Riemannian subspace limited-memory SR1 trust region method

Hejie Wei[1] · **Wei Hong Yang**[1]

Received: 30 July 2015 / Accepted: 23 November 2015 / Published online: 21 December 2015

Abstract In this paper, we present a new trust region algorithm on any compact Riemannian manifolds using subspace techniques. The global convergence of the method is proved and local $d+1$-step superlinear convergence of the algorithm is presented, where d is the dimension of the Riemannian manifold. Our numerical results show that the proposed subspace algorithm is competitive to some recent developed methods, such as the LRTR-SR1 method, the LRTR-BFGS method, the Riemannian CG method.

Keywords Subspace method · Riemannian optimization · LRTR-SR1 method · Vector transport

1 Introduction

Optimization on Riemannian manifolds (or Riemannian optimization) refers to the minimization of a real-valued function f defined on a Riemannian manifold $\mathcal{M}$, i.e.,

$$\begin{aligned} &\min f(x) \\ &\text{s.t.} \quad x \in \mathcal{M} \end{aligned} \tag{1.1}$$

✉ Hejie Wei
12110180023@fudan.edu.cn

Wei Hong Yang
whyang@fudan.edu.cn

1 School of Mathematical Sciences, Fudan University, 220 Handan Road, Shanghai 200433, China

where $\mathcal{M}$ is a Riemannian manifold. When $\mathcal{M}$ is the Euclidean space $\mathbb{R}^n$, (1.1) is just the traditional unconstrained optimization problem. Classical optimization algorithms on $\mathbb{R}^n$, such as steepest descent, Newton, quasi-Newton, nonlinear conjugate gradient and trust-region method have been successfully generalized to Riemannian manifolds. See [1–6] and the references therein. The reader refers to the toolbox 'manopt' in [7] for implementations of classical Riemannian optimization algorithms. For a particular manifold, there are efficient methods which exploit the special structure of the underlying manifold. See [16,19]. In this paper, we aim at proposing an algorithm for optimization problems on general Riemannian manifolds.

The Riemannian trust region (RTR) method proposed in [2] usually converges to a solution with high accuracy. However, the RTR method requires a function which computes the product of the Riemannian Hessian of f with a given tangent vector. It has been pointed out in [12] that the RTR framework seems to lose its appeal if the Hessian information of the objective function is unknown. For this reason, Huang has extended the limited memory SR1 (LSR1) trust region method on $\mathbb{R}^n$ to the Riemannian manifolds in [12]. Moreover, the local $d+1$ step superlinear convergence, where d is the dimension of $\mathcal{M}$, is also proved. Numerical experiments demonstrate that the SR1 trust region algorithm for the large scale problem is an effective method.

In the Euclidean case, the trust region method can be accelerated by subspace techniques for large scale problems. See [9–11,20,21] and the references therein. To the author's best knowledge, there have been few studies on subspace techniques for Riemannian optimization problems in the literature. This leads us to address issues relating LSR1 trust region method to subspace techniques.

In this paper, we present a subspace LSR1 trust region method on Rimennain manifolds. It is organized as follows: In Sect. 2, we introduce some notations and definitions that will be frequently used in the subsequent discussions. In Sect. 3, we present our subspace LSR1 trust region (SLRTR-SR1) method in detail. In Sect. 4, the global convergence and local $d+1$-step superlinear convergence are established. Finally, numerical tests are carried out in Sect. 5. Numerical results show the robustness and the efficiency of the new method compared with the LRTR-SR1 method, the LRTR-BFGS method and the Riemannian CG method.

2 Notations and definitions

In this section, we introduce the notations and definitions which will be used throughout the paper. For a matrix $A = (a_{ij})_{n\times n}$, $\mathrm{tr}(A) := \sum_{i=1}^{n} a_{ii}$ denotes the trace of A. The Frobenius norm of a $n \times p$ matrix B is defined by $\|B\|_F := \sqrt{\mathrm{tr}(B^T B)}$. If no confusion, the subscript F is omitted.

We use $\mathcal{M}$ to denote a Riemannian manifold. Given a point $x \in \mathcal{M}$, $T_x\mathcal{M}$ denotes the tangent space of $\mathcal{M}$ at x. The inner product defined on $T_x\mathcal{M}$ is denoted by $\langle\cdot,\cdot\rangle_x$ which induces a norm $\|\xi_x\|_x := \sqrt{\langle\xi_x, \xi_x\rangle}$ on $T_x\mathcal{M}$ (The subscript x may be omitted if there is no risk of confusion). The tangent bundle $T\mathcal{M} := \cup_x T_x\mathcal{M}$ consists of all tangent vectors to $\mathcal{M}$.

For a differentiable function $f : \mathcal{M} \to \mathbb{R}$, the derivative of f at $x \in \mathcal{M}$, denoted by $\mathrm{D}f(x)$, is an element of the dual space to $T_x\mathcal{M}$ which satisfies $\mathrm{D}f(x)v = vf$ for all $v \in T_x\mathcal{M}$. The gradient of f at $x \in \mathcal{M}$ (see [4]), denoted by $\mathrm{grad} f(x)$, is defined by

$$\langle \mathrm{grad} f(x), v\rangle = \mathrm{D} f(x)[v], \quad \forall v \in T_x\mathcal{M}.$$

For any $h \in T_x\mathcal{M}$, we use $h^\flat$ to denote the linear function on $T_x\mathcal{M}$ induced by h, that is,

$$h^\flat \eta = \langle h, \eta\rangle, \quad \forall \eta \in T_x\mathcal{M}. \tag{2.1}$$

Let $H = [h_1, \ldots, h_m]$, where $h_i \in T_x\mathcal{M}$ for any $i = 1, \ldots, m$. For $u \in \mathbb{R}^m$, $Hu = \sum_{i=1}^m u_i h_i$. Then H is a linear operator from $\mathbb{R}^m$ to $T_x\mathcal{M}$. Define $H^\flat : T_x\mathcal{M} \to \mathbb{R}^m$ by

$$H^\flat \eta = [h_1^\flat \eta, \ldots, h_m^\flat \eta]^T, \quad \forall \eta \in T_x\mathcal{M}. \tag{2.2}$$

For $x, y \in \mathcal{M}$, the Riemannian distance of x and y is denoted by $\mathrm{dist}(x, y)$ [4, p.46]. Let $B_\delta(x)$ be the set $\{y \in \mathcal{M} : \mathrm{dist}(x, y) < \delta\}$. We use **id** to denote the identity mapping on $T_x\mathcal{M}$.

Definition 21 [4, p. 55] A retraction on a manifold $\mathcal{M}$ is a smooth mapping R from the tangent bundle $T\mathcal{M}$ onto $\mathcal{M}$ with the following properties. Let R_x denote the restriction of R to $T_x\mathcal{M}$.

1. $R_x(0_x) = x$, where 0_x denotes the zero element of $T_x\mathcal{M}$.
2. With the canonical identification $T_{0x}(T_x\mathcal{M}) \simeq T_x\mathcal{M}$, R_x satisfies

$$\mathrm{D}R_x(0_x) = \mathbf{id}.$$

For a smooth function f and a corresponding retraction R_x, define the composite map

$$f_{R_x} = f \circ R_x : T_x\mathcal{M} \to \mathbb{R}. \tag{2.3}$$

Then $\mathrm{D}f_{R_x}(0) = \mathrm{D}f(x)$. We use $\mathrm{D}^2 f_{R_x}$ to denote the Hessian of f_{R_x}.

Assume that f is twice differentiable. The Riemannian Hessian of f at $x \in \mathcal{M}$ is denoted by $\mathrm{Hess} f(x)$. See [4, Definition 5.5.1] for the precise definition. Let x^* be a critical point of f ($\mathrm{grad} f(x^*) = 0$). From [4, Proposition 5.5.6], we know that

$$\mathrm{Hess} f(x^*) = \mathrm{Hess} f_{R_{x^*}}(0) = \mathrm{D}^2 f_{R_{x^*}}(0). \tag{2.4}$$

Definition 22 [4, Chapter 8] We will consider the transport of a vector from one tangent space $T_x\mathcal{M}$ into another one $T_y\mathcal{M}$, that is, consider isomorphisms $\mathcal{T}_{x,y} : T_x\mathcal{M} \longrightarrow T_y\mathcal{M}$. For a retraction R_x, the vector transport $\mathcal{T}_{x,y}^{R_x}$ is defined by

$$\mathcal{T}_{x,y}^{R_x} u := \mathrm{D}R_x(v)[u], \qquad \forall u \in T_x\mathcal{M},$$

i.e.

$$\mathcal{T}_{x,y}^{R_x} u = \frac{\mathrm{d}}{\mathrm{d}t} R_x(v + tu)\Big|_{t=0}, \qquad \forall u \in T_x\mathcal{M},$$

where $v = R_x^{-1}(y)$.

For simplicity, we omit the subscript R_x and use $\mathcal{T}_{x,y}$ to denote $\mathcal{T}_{x,y}^{R_x}$. Let $\mathcal{T}_{x,y}^*$ be the adjoint of $\mathcal{T}_{x,y}$ (defined by $\langle v, \mathcal{T}_{x,y}u\rangle = \langle \mathcal{T}_{x,y}^* v, u\rangle$ for all $u \in T_x\mathcal{M}$, $v \in T_y\mathcal{M}$). Then $\mathcal{T}_{x,y}$ is isometric under the condition $\mathcal{T}_{x,y} = \mathcal{T}_{y,x}^* = \mathcal{T}_{y,x}^{-1}$.

Given a smooth curve $\gamma : [a, b] \to \mathcal{M}$. The parallel translation along γ is denoted by $P_\gamma^{b\leftarrow a}$ (for precise definition, see [4, p. 105]). Parallel translation is a special vector transport. Furthermore, it is isometric.

Definition 23 [4, Definition 7.4.1] Let $f_{R_x} : T_x\mathcal{M} \to \mathbb{R}$ be defined by (2.3). We say that f_{R_x} is radially Lipschitz continuously differentiable if there exist $\beta > 0$ and $\delta > 0$ such that, for all $x \in \mathcal{M}$, for all $\xi \in T_x\mathcal{M}$ with $\|\xi\| = 1$, and for all $t < \delta$, it holds that

$$\left|\frac{\mathrm{d}}{\mathrm{d}\tau} f_{R_x}(\tau\xi)\Big|_{\tau=t} - \frac{\mathrm{d}}{\mathrm{d}\tau} f_{R_x}(\tau\xi)\Big|_{\tau=0}\right| \le \beta t. \tag{2.5}$$

3 The Riemannian subspace method

Throughout the paper, the gradient of f at iterate x_k is denoted by g_k. Given a current iterate x_k, the general trust region methods construct a quadratic model m_k as follows:

$$m_k(s) := f(x_k) + \langle g_k, s\rangle + \frac{1}{2}\langle \mathcal{B}_k s, s\rangle, \qquad \forall s \in T_{x_k}\mathcal{M},$$

where $\mathcal{B}_k : T_{x_k}\mathcal{M} \to T_{x_k}\mathcal{M}$ is a symmetric linear operator. The approximate solution of the trust region subproblem

$$\begin{aligned} \min_{s\in T_{x_k}\mathcal{M}} \quad & m_k(s) \\ \text{s.t.} \quad & \|s\| \le \Delta_k \end{aligned} \tag{3.1}$$

is denoted by s_k. Let ρ denote the quotient of the actual decrease and the predicted decrease, that is,

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

In the RTR method, if ρ is greater than a preassigned value, the new iterate $x_{k+1} = R_{x_k}(s_k)$ is accepted and the trust-region radius Δ_k is updated. See Algorithm 10 in [4, Chapter 7]) for detail.

3.1 The LRTR-SR1 method

The Riemannian limited-memory SR1 trust region method (abbreviated as LRTR-SR1) is introduced by Huang in [12]. It relies on the Riemannian generalization of the compact representation of the classic SR1 matrices presented in [8,13]. For $k \geq 0$, let

$$s_k = R_{x_k}^{-1}(x_{k+1}), \quad y_k = \mathcal{T}_{x_k,x_{k+1}}^{-1} g_{k+1} - g_k. \tag{3.2}$$

It is obvious that $s_k, y_k \in T_{x_k}\mathcal{M}$. For a non-negative integer $i < k$, we use the following notations:

$$s_i^{(k)} = \mathcal{T}_{x_i,x_k} s_i, \quad y_i^{(k)} = \mathcal{T}_{x_i,x_k} y_i.$$

which transport s_i, y_i in $T_{x_i}\mathcal{M}$ into $s_i^{(k)}, y_i^{(k)}$ in $T_{x_k}\mathcal{M}$.

For a positive integer m, as in the Euclidean case, we use $S_{k,m}$ and $Y_{k,m}$ to denote the m most recent corrections of s_i and y_i. In the Riemannian setting, all these m most recent corrections must be in the tangent space $T_{x_k}\mathcal{M}$, which yields

$$S_{k,m} = \left\{s_{k-l}^{(k)}, s_{k-l+1}^{(k)}, \ldots, s_{k-1}^{(k)}\right\}, \quad Y_{k,m} = \left\{y_{k-l}^{(k)}, y_{k-l+1}^{(k)}, \ldots, y_{k-1}^{(k)}\right\}, \tag{3.3}$$

where $l = \min\{m, k\}$.

Now we present the Riemannian generalization of compact form of the SR1 formula (see [13, (7.30)]). Let $D_{k,m}$, $L_{k,m}$ be defined as the following form

$$D_{k,m} = \operatorname{diag}[\langle s_{k-l}, y_{k-l}\rangle, \langle s_{k-l+1}, y_{k-l+1}\rangle, \ldots, \langle s_{k-1}, y_{k-1}\rangle], \tag{3.4}$$

$$L_{k,m} = \begin{cases} \langle s_{k-l+i-1}^{(k)}, y_{k-l+j-1}^{(k)}\rangle, & \text{if } i > j, \\ 0, & \text{otherwise}. \end{cases}, \tag{3.5}$$

and let $P_{k,m} = D_{k,m} + L_{k,m} + L_{k,m}^T$. The Riemannian Hessian approximation $\mathcal{B}_k$ is defined by

$$\mathcal{B}_k = \mathcal{B}_0^k + \left(Y_{k,m} - \mathcal{B}_0^k S_{k,m}\right)\left(P_{k,m} - S_{k,m}^{\flat}\mathcal{B}_0^k S_{k,m}\right)^{-1}\left(Y_{k,m} - \mathcal{B}_0^k S_{k,m}\right)^{\flat}, \quad k > 0, \tag{3.6}$$

where $\mathcal{B}_0^k = \gamma_k \mathbf{id}$, $\gamma_k = \dfrac{\langle y_{k-1}, y_{k-1}\rangle}{\langle s_{k-1}, y_{k-1}\rangle}$.

The trust region method with the limited memory SR1 update (3.6) is called the LRTR-SR1 method. For theory and implementation, the reader refers to [12].

3.2 The subspace LRTR-SR1 method

In this section, we study the subspace properties of the trial step s_k, which is the solution of the LRTR-SR1 subproblem (3.1) at the k-th iteration. Recall that for the

LRTR-SR1 subproblem, $\mathcal{B}_k$ in (3.1) is defined by (3.6). If we solve (3.1) with the truncated CG method, then

$$s_k \in \operatorname{span}\left\{g_k, \mathcal{B}_k g_k, \dots, \mathcal{B}_k^j g_k\right\},$$

where j stands for the inner iteration number of solving the subproblem at x_k. From (3.3) and (3.6), it follows that

$$s_k \in \mathcal{S}_k := \operatorname{span}\left\{g_k, s_{k-l}^{(k)}, \dots, s_{k-1}^{(k)}, y_{k-l}^{(k)}, \dots, y_{k-1}^{(k)}\right\},$$

where $l = \min\{m, k\}$. It is obvious that solving the trust region subproblem within the subspace $\mathcal{S}_k$ may generate a better decent direction than s_k.

Moveover, since the dimension of $\mathcal{S}_k$ is usually very small, the subspace subproblem can be solved exactly with cheap cost. Thus, the subspace method will be suitable for large-scale problems.

To avoid numerical problems, as in [18], define A_k as

$$A_k = \left[\frac{g_k}{\|g_k\|}, \frac{s_{k-l}^{(k)}}{\|s_{k-l}^{(k)}\|}, \dots, \frac{s_{k-1}^{(k)}}{\|s_{k-1}^{(k)}\|}, \frac{y_{k-l}^{(k)}}{\|y_{k-l}^{(k)}\|}, \dots, \frac{y_{k-1}^{(k)}}{\|y_{k-1}^{(k)}\|}\right]. \tag{3.7}$$

Then for any $s \in \mathcal{S}_k$, $s = A_k z$ for some $z \in \mathbb{R}^{2l+1}$. The subspace trust region subproblem of our algorithm is defined as follows:

$$\min_{z\in\mathbb{R}^{2l+1}} \bar{m}_k(z) = f(x_k) + \langle g_k, A_k z\rangle + \frac{1}{2}\langle \mathcal{B}_k A_k z, A_k z\rangle, \text{ s.t. } \|z\|_2 \le \Delta_k,$$

which can be equivalently written as

$$\min_{z\in\mathbb{R}^{2l+1}} \bar{m}_k(z) = f(x_k) + \langle \bar{g}_k, z\rangle + \frac{1}{2}\langle \bar{B}_k z, z\rangle, \text{ s.t. } \|z\|_2 \le \Delta_k, \tag{3.8}$$

where

$$\bar{g}_k = A_k^\flat g_k = \left[\frac{\langle g_k, g_k\rangle}{\|g_k\|}, \frac{\langle s_{k-l}^{(k)}, g_k\rangle}{\|s_{k-l}^{(k)}\|}, \dots, \frac{\langle s_{k-1}^{(k)}, g_k\rangle}{\|s_{k-1}^{(k)}\|}, \frac{\langle y_{k-l}^{(k)}, g_k\rangle}{\|y_{k-l}^{(k)}\|}, \dots, \frac{\langle y_{k-1}^{(k)}, g_k\rangle}{\|y_{k-1}^{(k)}\|}\right]^T, \tag{3.9}$$

and

$$\bar{B}_k = A_k^\flat \mathcal{B}_k A_k. \tag{3.10}$$

By (3.4), during the first m iterations, after the new iterate x_{k+1} is generated, we obtain $D_{k+1,m}$ by adding the new element $\langle s_k, y_k\rangle$ from $D_{k,m}$. At subsequent iterations $k > m$, we obtain $D_{k+1,m}$ by deleting $\langle s_{k-m}, y_{k-m}\rangle$ from $D_{k,m}$ and adding the new displacement $\langle s_k, y_k\rangle$. And by (3.5), we update $L_{k+1,m}$ in a similar fashion. Therefore, it is easy to update $P_{k,m}$ based on its definition.

The middle matrix $P_{k,m} - \gamma_k S^\flat_{k,m} S_{k,m}$ is small–of dimension l by (3.6). Thus it is easy to obtain its orthogonal factorization. Let J be the orthogonal matrix such that $P_{k,m} - \gamma_k S^\flat_{k,m} S_{k,m} = J^T \Lambda J$, where Λ is diagonal. Then

$$\begin{aligned}
\bar{B}_k &= A_k^\flat \mathcal{B}_k A_k \\
&= \gamma_k A_k^\flat A_k + A_k^\flat (Y_{k,m} - \gamma_k S_{k,m}) \left(P_{k,m} - \gamma_k S^\flat_{k,m} S_{k,m}\right)^{-1} (Y_{k,m} - \gamma_k S_{k,m})^\flat A_k \\
&= \gamma_k A_k^\flat A_k + A_k^\flat (Y_{k,m} - \gamma_k S_{k,m}) J^T \Lambda^{-1} J (Y_{k,m} - \gamma_k S_{k,m})^\flat A_k \\
&= \gamma_k A_k^\flat A_k + W^T \Lambda^{-1} W, \qquad (3.11)
\end{aligned}$$

where $W = J(Y_{k,m} - \gamma_k S_{k,m})^\flat A_k$.

The equation (3.11) indicates that updating the matrix $\bar{B}_k$ is quite economical when m is small. Since the subproblem (3.8) is of small size (dimension $2l + 1$), solving it requires a negligible amount of computation. Therefore, the cost of each loop iteration is very small.

The subspace limited memory SR1 trust region algorithm can be stated formally as follows (see Algorithm 1).

Algorithm 1 Subspace LSR1 trust region optimization method (SLRTR-SR1 method)
Require: Riemannian manifold $\mathcal{M}$; isometric vector transport $\mathcal{T}$ on $\mathcal{M}$ with associated
retraction R; real-valued function f on $\mathcal{M}$.
Goal: Find a minimizer of f.
Parameters: $\epsilon \geq 0, 0 < \tau_1 < \tau_2 < 1, 0 < c_1 < 1, c_2 > 1, m \geq 0, l = \min\{m, k\}$.
Input: Initial iterate $x_0 \in \mathcal{M}$.
Output: Stationary point within the given precision.

```
Step 1: k=0; set $g_0 = -\text{grad}\, f(x_0)$, $\gamma_0 > 0$, $\Delta_0 > 0$, $Y_0 = S_0 = \mathbf{0}$, $A_0 = [\frac{g_0}{\|g_0\|}]$;
Step 2: Solve the subspace trust region subproblem (3.8), that is,
            $z_k = \text{argmin}_{z \in \mathbb{R}^{2l+1}} \bar{m}_k(z) = f(x_k) + \langle \bar{g}_k, z \rangle + \frac{1}{2} \langle \bar{B}_k z, z \rangle, s.t. \|z\|_2 \leq \Delta_k$
        to get $z_k$ where $\bar{g}_k, \bar{B}_k$ is defined as (3.9), (3.11) , and set $s_k = A_k z_k$;
Step 3: Compute $\bar{m}_k(z_k)$ and $\rho_k = \frac{f(x_k) - f(R_{x_k}(s_k))}{\bar{m}_k(0) - \bar{m}_k(z_k)}$;
Step 4: if $\rho_k < \tau_1$ then $\Delta_k = c_1 \Delta_k$ and go to Step 2;
Step 5: else if $\rho_k > \tau_2$ then $\Delta_{k+1} = c_2 \Delta_k$ end if;
Step 6: else $\Delta_{k+1} = \Delta_k$ end if;
Step 7: Compute $x_{k+1} = R_{x_k}(s_k)$, $y_k = \mathcal{T}^{-1}_{k,k+1} g_{k+1} - g_k$ and $\gamma_k = \frac{\langle y_k, y_k \rangle}{\langle s_k, y_k \rangle}$ ;
Step 8: Transport $s_{k-l+1}, \ldots, s_{k-1}, s_k, y_{k-l+1}, \ldots, y_{k-1}, y_k$ from $T_{x_k}\mathcal{M}$ to $T_{x_{k+1}}\mathcal{M}$ and
        update the correction sets $Y_{k+1,m}, S_{k+1,m}$;
Step 9: $k \leftarrow k + 1$, and calculate $A_k, \bar{g}_k, \bar{B}_k$ by (3.7), (3.9), (3.11);
        go to Step 2 until convergence.
```

4 Convergence analysis

4.1 Global convergence

In this subsection, we show that our SLRTR-SR1 algorithm is globally convergent. Before establishing the main results, we make an assumption.

Assumption 41 Every $\mathcal{B}_k$ in Algorithm 1 satisfies $\|\mathcal{B}_k\| \leq M$ for a uniform constant M.

Assumption 42 Every A_k in Algorithm 1 is full rank.

Theorem 41 *Let $\{x_k\}$ be the sequence of iterates generated by Algorithm 1. Suppose that f is continuous differentiable and bounded below on the level set $\{x \in \mathcal{M} : f(x) < f(x_0)\}$, that f_{R_x} is radially Lipschitz continuously differentiable, and that all $\mathcal{B}_k$ satisfy the Assumption 41. Then we have*

$$\liminf_{k\to\infty} \|\text{grad } f(x_k)\| = 0.$$

Proof By Theorem 4 in [13], since z_k is the solution of (3.8), we have

$$\bar{m}_k(0) - \bar{m}_k(z_k) \geq \frac{1}{2}\|\bar{g}_k\|_2 \min\left\{\Delta_k, \frac{\|\bar{g}_k\|_2}{\|\bar{B}_k\|}\right\}. \tag{4.1}$$

It is obvious by (3.7) that $\|A_k\| = \sqrt{2l+1}$. From Assumption 41, (3.7) and (3.9), it follows that

$$\begin{aligned} \|\bar{g}_k\|_2 &= \|A_k^\flat g_k\|_2 \geq \|g_k\|, \\ \|\bar{B}_k\| &= \|A_k^\flat \mathcal{B}_k A_k\| \leq (2l+1)M. \end{aligned}$$

Assume for contraction that there exist $\epsilon > 0$ and K such that

$$\|g_k\| \geq \epsilon, \text{ for all } k \geq K,$$

which together with (4.1) implies

$$\bar{m}_k(0) - \bar{m}_k(z_k) \geq \frac{\epsilon}{2}\min\left\{\Delta_k, \frac{\epsilon}{(2l+1)M}\right\}. \tag{4.2}$$

Since the sequence $\{f(x_k)\}$ decrease monotonically and f is bounded below, $f(x_k)$ converges to f^* for some scalar f^*. By our algorithm, we have

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= f(x_k) - f(R_{x_k}(A_k z_k)) \\ &\geq \tau_1(\bar{m}_k(0) - \bar{m}_k(z_k)) \\ &\geq \frac{\epsilon\tau_1}{2}\min\left\{\Delta_k, \frac{\epsilon}{(2l+1)M}\right\}, \end{aligned} \tag{4.3}$$

where the last inequality follows from (4.2).

Letting $k \to \infty$ in (4.3), we get $\Delta_k \to 0$. Then there exists a positive index $k_1 \geq K$ such that

$$\bar{m}_k(0) - \bar{m}_k(z_k) \geq \frac{1}{2}\epsilon \Delta_k, \text{ for all } k \geq k_1. \tag{4.4}$$

Recall that f_{R_x} is defined by (2.3). For any $\eta_k \in T_{x_k}\mathcal{M}$, by the first-order Taylor expansion of $f_{R_{x_k}}(\eta_k)$ and (2.5), we have

$$\begin{aligned} f_{R_{x_k}}(\eta_k) &= f_{R_{x_k}}(0) + \|\eta_k\| \frac{\mathrm{d}}{\mathrm{d}\tau} f_{R_{x_k}}\left(\tau \frac{\eta_k}{\|\eta_k\|}\right)\Big|_{\tau=0} + r_k \\ &= f(x_k) + \langle g_k, \eta_k \rangle + r_k, \end{aligned} \tag{4.5}$$

where $|r_k| \leq \frac{1}{2}\beta\|\eta_k\|^2$. It follows from (3.8) and (4.5) that

$$\begin{aligned} |\bar{m}_k(z_k) - f_{R_{x_k}}(A_k z_k)| &= \Big|\frac{1}{2}\langle \bar{B}_k z_k, z_k\rangle - (f_{R_{x_k}}(A_k z_k) - f(x_k) - \langle g_k, A_k z_k\rangle)\Big| \\ &\leq \frac{1}{2}(2l+1)M\|z_k\|_2^2 + \frac{1}{2}\beta\|A_k z_k\|^2 \\ &\leq \tilde{M}\|z_k\|_2^2, \end{aligned} \tag{4.6}$$

where $\tilde{M} = \max\left\{\frac{1}{2}(2l+1)M, \ \frac{1}{2}(2l+1)\beta\right\}$. By the definition of the ratio ρ_k, (4.4) and (4.6), we have

$$\begin{aligned} |\rho_k - 1| &= \left|\frac{\bar{m}_k(z_k) - f_{R_{x_k}}(A_k z_k)}{\bar{m}_k(0) - \bar{m}_k(z_k)}\right| \\ &\leq \frac{\tilde{M}\|z_k\|_2^2}{\epsilon\Delta_k/2} \leq \frac{2\tilde{M}\Delta_k}{\epsilon}. \end{aligned}$$

Since $\Delta_k \to 0$, we have $\rho_k \to 1$. Then there exists a $k_2 > k_1$, such that $\rho_k > \tau_1$ for all $k \geq k_2$. By Step 5 and Step 6 of our algorithm, there exists a $\bar{\Delta} > 0$ such that $\Delta_k \geq \bar{\Delta}$ for all $k \geq k_2$, which contradicts $\Delta_k \to 0$. Hence our assertion holds. □

4.2 Local convergence

We use d to denote the dimension of $\mathcal{M}$, that is, $d = \dim\mathcal{M}$. Under some mild conditions, the local $d+1$-step superlinear convergence of our method can be established. To obtain the local convergence, assume that f is twice continuous differentiable and

$$x_k \textbf{ converges to } x^*, \tag{4.7}$$

where x^* is a local minimizer of f. We also make the following additional assumptions:

Assumption 43 Hess $f(x^*)$ is positive definite.

Assumption 44 There exist constants c_0, δ such that for all $x, y \in B_\delta(x^*)$,

$$\|\mathrm{Hess} f(y) - \mathcal{T}_{x,y}\mathrm{Hess}\ f(x)\mathcal{T}_{x,y}^{-1}\| \le c_0 \mathrm{dist}(x, y). \tag{4.8}$$

By Assumption 43 and (4.7), if k becomes sufficiently large, then $\mathrm{Hess} f(x_k)$ is positive definite and satisfies

$$\|\mathrm{Hess} f(x_k)^{-1}\| \le b_0 \tag{4.9}$$

for some $b_0 > 0$. In the following, we give some preparing results.

Lemma 41 [12, Lemma 3.3.2] *Let $\mathcal{M}$ be a Riemannian manifold endowed with a retraction R and let $\bar{x} \in \mathcal{M}$. Then there exists $b_1, b_2, \bar{\delta}_{b_1,b_2} > 0$ such that for all x in a small neighborhood of $\bar{x}$ and all $\xi, \eta \in \mathcal{T}_x\mathcal{M}$ with $\|\xi\| \le \bar{\delta}_{b_1,b_2}$ and $\|\eta\| \le \bar{\delta}_{b_1,b_2}$,*

$$b_1\|\xi - \eta\| \le \mathrm{dist}(R_x(\xi), R_x(\eta)) \le b_2\|\xi - \eta\|. \tag{4.10}$$

Letting $\eta = 0$ in (4.10), we obtain the following inequality, which will be used later,

$$b_1\|\xi\| \le \mathrm{dist}(x, R_x(\xi)) \le b_2\|\xi\|. \tag{4.11}$$

Let $\lambda_{\max}$ and $\lambda_{\min}$ be the maximal and minimal eigenvalues of $\mathrm{Hess} f(x^*)$.

Lemma 42 [2, Lemma 4.8] *Given $\lambda_1 < \lambda_{min}$ and $\lambda_2 > \lambda_{max}$, there exists a neighborhood $\mathcal{U}$ of x^* such that*

$$\lambda_1 \mathrm{dist}(x, x^*) \le \|\mathrm{grad}\ f(x)\| \le \lambda_2 \mathrm{dist}(x, x^*), \qquad \forall x \in \mathcal{U}.$$

Let F and G be two functions defined on a neighborhood $\mathcal{N}$ of x^*. We say that $F(x) = \Omega(G(x))$ for all $x \in \mathcal{N}$, which means that there exist $c, C > 0$ such that $c|F(x)| \le |G(x)| \le C|F(x)|$ for all $x \in \mathcal{N}$.

Lemma 43 *The following statement holds*

$$f(x) - f(x^*) = \Omega(dist^{\,2}(x, x^*))$$

for all x in a small neighborhood of x^*.

Proof By the Definition 21, there exists a neighborhood $\mathcal{W}$ of x^* such that $R_{x^*}^{-1}$ exists on $\mathcal{W}$. Then, for all $x \in \mathcal{W}$,

$$\begin{aligned} f(x) - f(x^*) &= f_{R_{x^*}}\left(R_{x^*}^{-1}(x)\right) - f_{R_{x^*}}(0_{x^*}) \\ &= \mathrm{D}f(x^*)R_{x^*}^{-1}(x) + \frac{1}{2}\mathrm{D}^2 f_{R_{x^*}}\left(tR_{x^*}^{-1}(x)\right)\left(R_{x^*}^{-1}(x), R_{x^*}^{-1}(x)\right) \end{aligned}$$

for some $t \in (0, 1)$. Since x^* is a local minimizer, we have $\mathrm{grad} f(x^*) = 0$ and therefore

$$f(x) - f(x^*) = \frac{1}{2}\mathrm{D}^2 f_{R_{x^*}}\left(tR_{x^*}^{-1}(x)\right)\left(R_{x^*}^{-1}(x), R_{x^*}^{-1}(x)\right). \tag{4.12}$$

From Assumption 43 and (2.4), we know that $\mathrm{D}^2 f_{R_{x^*}}(0)$ is positive definite, which implies that $\mathrm{D}^2 f_{R_{x^*}}$ is positive definite and bounded on a neighborhood $\mathcal{V}$ of x^*. Let $\mathcal{N} = \mathcal{V} \cap \mathcal{W}$. For all $x \in \mathcal{N}$, from (4.12), it is easy to see that

$$f(x) - f(x^*) = \Omega\left(\|R_{x^*}^{-1}(x)\|^2\right). \tag{4.13}$$

From the proof of [4, Prop. 7.1.3], we know that $\|R_{x^*}^{-1}(x)\| = \Omega(\mathrm{dist}(x, x^*))$, which together with (4.13) implies the assertion. □

Suppose that $\mathcal{B}_k, s_{k-l}^{(k)}, s_{k-l+1}^{(k)}, \ldots, s_{k-1}^{(k)}$ and $y_{k-l}^{(k)}, y_{k-l+1}^{(k)}, \ldots, y_{k-1}^{(k)}$ are generated by Algorithm 1. Next we show that the secant condition holds for $\mathcal{B}_k$.

Lemma 44 *For all $k \geq 1$, we have*

$$\mathcal{B}_k s_{k-1}^{(k)} = y_{k-1}^{(k)}.$$

Proof From (3.6), it follows that

$$\mathcal{B}_k s_{k-1}^{(k)} = \gamma_k s_{k-1}^{(k)} + (Y_{k,m} - \gamma_k S_{k,m})\left(P_{k,m} - \gamma_k S_{k,m}^{\flat} S_{k,m}\right)^{-1} (Y_{k,m} - \gamma_k S_{k,m})^{\flat} s_{k-1}^{(k)}. \tag{4.14}$$

By the definitions of $P_{k,m}$, $S_{k,m}$ and $Y_{k,m}$, we have

$$\begin{aligned} &P_{k,m} - \gamma_k S_{k,m}^{\flat} S_{k,m} \\ &= \begin{bmatrix} \langle y_{k-l}^{(k)} - \gamma_k s_{k-l}^{(k)}, s_{k-l}^{(k)} \rangle & \langle y_{k-l}^{(k)} - \gamma_k s_{k-l}^{(k)}, s_{k-l+1}^{(k)} \rangle & \cdots & \langle y_{k-l}^{(k)} - \gamma_k s_{k-l}^{(k)}, s_{k-1}^{(k)} \rangle \\ \langle y_{k-l}^{(k)} - \gamma_k s_{k-l}^{(k)}, s_{k-l+1}^{(k)} \rangle & \langle y_{k-l+1}^{(k)} - \gamma_k s_{k-l+1}^{(k)}, s_{k-l+1}^{(k)} \rangle & \cdots & \langle y_{k-l+1}^{(k)} - \gamma_k s_{k-l+1}^{(k)}, s_{k-1}^{(k)} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle y_{k-l}^{(k)} - \gamma_k s_{k-l}^{(k)}, s_{k-1}^{(k)} \rangle & \langle y_{k-l+1}^{(k)} - \gamma_k s_{k-l+1}^{(k)}, s_{k-1}^{(k)} \rangle & \cdots & \langle y_{k-1}^{(k)} - \gamma_k s_{k-1}^{(k)}, s_{k-1}^{(k)} \rangle \end{bmatrix}, \end{aligned}$$

and

$$(Y_{k,m} - \gamma_k S_{k,m})^{\flat} s_{k-1}^{(k)} = \begin{pmatrix} \langle y_{k-l}^{(k)} - \gamma_k s_{k-l}^{(k)}, s_{k-1}^{(k)} \rangle \\ \langle y_{k-l+1}^{(k)} - \gamma_k s_{k-l+1}^{(k)}, s_{k-1}^{(k)} \rangle \\ \vdots \\ \langle y_{k-1}^{(k)} - \gamma_k s_{k-1}^{(k)}, s_{k-1}^{(k)} \rangle \end{pmatrix},$$

which is just the last column of $P_{k,m} - \gamma_k S_{k,m}^{\flat} S_{k,m}$. Thus we have

$$\left(P_{k,m} - \gamma_k S_{k,m}^{\flat} S_{k,m}\right)^{-1} (Y_{k,m} - \gamma_k S_{k,m})^{\flat} s_{k-1}^{(k)} = [0, 0, \ldots, 1]^T,$$

and therefore

$$(Y_{k,m} - \gamma_k S_{k,m})\left(P_{k,m} - \gamma_k S_{k,m}^{\flat} S_{k,m}\right)^{-1} (Y_{k,m} - \gamma_k S_{k,m})^{\flat} s_{k-1}^{(k)} = y_{k-1}^{(k)} - \gamma_k s_{k-1}^{(k)}.$$

Then our assertion follows from the above equality and (4.14). □

For $k \geq 1$, let $e_k := \text{dist}(x_k, x^*)$ and

$$H_k := \mathcal{T}_{x^*,x_k} \text{Hess} f(x^*) \mathcal{T}^{-1}_{x^*,x_k}.$$

Based on Lemma 44, we can prove the following result. Its proof is similar to [12, Lemma 3.3.13] and so we omit it.

Lemma 45 *Let p be an integer satisfying $p \geq d$. There exists N such that for any set of $p+1$ consecutive steps $s_k, s_{k+1}, s_{k+2}, \ldots, s_{k+p}$ with $k \geq N$, there exists an index set $\mathcal{G}_k$, of at least $p+1-d$ indices contained in the set $\{i : k+1 \leq i \leq k+p\}$ such that for all $j \in \mathcal{G}_k$,*

$$\frac{\|(\mathcal{B}_j - H_j)s_j\|}{\|s_j\|} < b_3 \epsilon_k^{1/d}$$

where $b_3 > 0$ is independent of $\mathcal{G}_k$, and

$$\epsilon_k := \max_{k+1 \leq j \leq k+p} \{e_j, \text{dist}(R_{x_j}(s_j), x^*)\}. \tag{4.15}$$

For fixed p, by (4.7) and (4.15), it is easy to prove that ϵ_k converges to 0 as k goes to infinity.

Lemma 46 *Assume that there exists $\theta > 0$ such that $\|\bar{B}_k z_k + \bar{g}_k\| \leq \|\bar{g}_k\|^{1+\theta}$ for all sufficiently large k. If k is sufficiently large, there exist $b_7, b_8 > 0$ such that*

$$\text{dist}(x_{k+1}, x^*) \leq b_7 \|(\text{Hess} f(x_k) - \mathcal{B}_k)s_k\| + b_8 e_k^{1+\min\{\theta,1\}}.$$

Proof Let $\xi_k = R^{-1}_{x_k}(x^*)$. Then $\xi_k \in T_{x_k}\mathcal{M}$. When k is sufficiently large, x_k is in a small neighborhood of x^*, and $\|s_k\|$, $\|\xi_k\|$ are sufficiently small. From Lemma 41, it follows that $\text{dist}(R_{x_k}(s_k), R_{x_k}(\xi_k)) \leq b_2\|s_k - \xi_k\|$. Note that $x_{k+1} = R_{x_k}(s_k)$ from (3.2), and therefore

$$\text{dist}(x_{k+1}, x^*) \leq b_2 \|s_k - \xi_k\|. \tag{4.16}$$

Let $\delta_k = \mathcal{B}_k s_k + g_k$. By (3.9) and (3.10), it is easy to see that $A_k(\bar{B}_k z_k + \bar{g}_k) = (2l+1)\delta_k$, which together with the assumption of the Lemma implies that

$$\|\delta_k\| \leq \frac{\|A_k\|}{2l+1}\|\bar{B}_k z_k + \bar{g}_k\| \leq \frac{\sqrt{2l+1}}{2l+1}\|A_k^\flat g_k\|^{1+\theta} \leq \|g_k\|^{1+\theta}.$$

For sufficiently large k, x_k satisfies the assumption of Lemma 42, and therefore, $\|g_k\| \leq \lambda_2 \text{dist}(x_k, x^*) = \lambda_2 e_k$ for some $\lambda_2 > 0$. Thus we have

$$\|\delta_k\| \leq \lambda_2^{1+\theta} e_k^{1+\theta}. \tag{4.17}$$

For sufficently large k, $\text{Hess} f(x_k)$ is postitive definite and bounded. Thus,

$$s_k = \text{Hess} f(x_k)^{-1}[(\text{Hess} f(x_k) - \mathcal{B}_k)s_k - g_k + \delta_k]. \tag{4.18}$$

Let γ be the curve defined by $\gamma(t) = R_{x_k}(t\xi_k), t \in [0, 1]$. Obviously, $\gamma(0) = x_k$ and $\gamma(1) = x^*$. By (4.18) and (4.9), we obtain

$$\begin{aligned}
\|s_k - \xi_k\| &= \|\mathrm{Hess} f(x_k)^{-1}\big[(\mathrm{Hess} f(x_k) - \mathcal{B}_k)s_k - g_k + \delta_k - \mathrm{Hess} f(x_k)\xi_k\big]\| \\
&\le b_0 \Bigg(\|(\mathrm{Hess} f(x_k) - \mathcal{B}_k)s_k\| + \Bigg\| P_\gamma^{0\leftarrow 1} g^* - g_k \\
&\quad - \left(\int_0^1 P_\gamma^{0\leftarrow t} \mathrm{Hess} f(\gamma(t)) \right) P_\gamma^{t\leftarrow 0} dt \Bigg) \xi_k \Bigg\| \\
&\quad + \left\| \Big(\int_0^1 P_\gamma^{0\leftarrow t} \mathrm{Hess} f(\gamma(t)) P_\gamma^{t\leftarrow 0} dt \Big) \xi_k - \mathrm{Hess} f(x_k)\xi_k \right\| + \|\delta_k\| \Bigg).
\end{aligned} \tag{4.19}$$

By [12, Lemma 3.3.8], there exists $b_4 > 0$ such that

$$\| P_\gamma^{0\leftarrow 1} g^* - g_k - \left(\int_0^1 P_\gamma^{0\leftarrow t} \mathrm{Hess} f(\gamma(t)) P_\gamma^{t\leftarrow 0} dt \right) \xi_k \| \le b_4 \|\xi_k\|^2. \tag{4.20}$$

From Assumption 44, there is $b_5 > 0$ such that

$$\left\| \left(\int_0^1 P_\gamma^{0\leftarrow t} \mathrm{Hess} f(\gamma(t)) P_\gamma^{t\leftarrow 0} dt \right) \xi_k - \mathrm{Hess} f(x_k)\xi_k \right\| \le b_5 \|\xi_k\|^2. \tag{4.21}$$

Combining (4.20), (4.21) with (4.19), and using (4.17), we have

$$\begin{aligned}
\|s_k - \xi_k\| &\le b_0 \left(\|(\mathrm{Hess} f(x_k) - \mathcal{B}_k)s_k\| + (b_4 + b_5)\|\xi_k\|^2 + \lambda_2^{1+\theta} e_k^{1+\theta} \right) \\
&\le b_0 \|(\mathrm{Hess} f(x_k) - \mathcal{B}_k)s_k\| + b_6 e_k^{1+\min\{\theta,1\}},
\end{aligned} \tag{4.22}$$

where the last inequality follows from (4.11) and $e_k < 1$. It follows from (4.22) and (4.16) that

$$\mathrm{dist}(x_{k+1}, x^*) \le b_7 \|(\mathrm{Hess} f(x_k) - \mathcal{B}_k)s_k\| + b_8 e_k^{1+\min\{\theta,1\}},$$

in which $b_7 = b_0 b_2$, $b_8 = b_2 b_6$. The proof is complete. □

Now we are in a position to prove the local $d + 1$-step superlinear convergence of our algorithm, which is another main result of this paper.

Theorem 42 *Assume that there exists $\theta > 0$ such that $\|\bar{B}_k z_k + \bar{g}_k\| \le \|\bar{g}_k\|^{1+\theta}$ for all sufficiently large k. Then the sequence $\{x_k\}$ generated by Algorithm 1 is locally $d + 1$ step superlinear convergent, that is,*

$$\lim_{k\to\infty} \frac{\mathrm{dist}(x_{k+d+1}, x^*)}{\mathrm{dist}(x_k, x^*)} = 0.$$

Proof By Lemma 45 (let $p = d$), there exists N such that if $k > N$, the set of steps $\{s_k, s_{k+1}, s_{k+2}, \dots, s_{k+d}\}$ contains at least one step s_j, $k+1 \le j \le k+d$, such that

$$\frac{\|(\mathcal{B}_j - H_j)s_j\|}{\|s_j\|} < b_3 \epsilon_k^{1/d} \tag{4.23}$$

for some $b_3 > 0$.

By (4.7), we can also assume that N is large enough such that the following statements hold:

(1) For all $i > N$, $\operatorname{dist}(x_i, x^*) < \delta$, where δ is as in Assumption 44.
(2) By Lemma 43, there exist λ_3, λ_4 such that for all $i > N$,

$$\lambda_3 \operatorname{dist}^2(x_i, x^*) \le f(x_i) - f(x^*) \le \lambda_4 \operatorname{dist}^2(x_i, x^*). \tag{4.24}$$

Since $j > N$, from the discussion above, we know that (4.8) holds for x_j and x^*. By the definition of H_j, we have

$$\begin{aligned}
&\frac{\|(\operatorname{Hess} f(x_j) - \mathcal{B}_j)s_j\|}{\|s_j\|} \\
&= \frac{\|(\operatorname{Hess} f(x_j) - \mathcal{T}_{x^*,x_j} \operatorname{Hess} f(x^*) \mathcal{T}_{x^*,x_j}^{-1} + \mathcal{T}_{x^*,x_j} \operatorname{Hess} f(x^*) \mathcal{T}_{x^*,x_j}^{-1} - \mathcal{B}_j)s_j\|}{\|s_j\|} \\
&\le \frac{\|\operatorname{Hess} f(x_j) - \mathcal{T}_{x^*,x_j} \operatorname{Hess} f(x^*) \mathcal{T}_{x^*,x_j}^{-1}\| \|s_j\|}{\|s_j\|} + \frac{\|(H_j - \mathcal{B}_j)s_j\|}{\|s_j\|} \\
&\le c_0 \operatorname{dist}(x_j, x^*) + b_3 \epsilon_k^{1/d} && \text{(by (4.8) and (4.23))} \\
&\le \tilde{c}_0 \sqrt{f(x_j) - f(x^*)} + b_3 \epsilon_k^{1/d} && \text{(by (4.24))} \\
&\le \tilde{c}_0 \sqrt{f(x_k) - f(x^*)} + b_3 \epsilon_k^{1/d} && (f(j) \le f(x_k) \text{due to} j \ge k+1) \\
&\le b_9 \epsilon_k^{1/d} && \text{(by (4.24))}
\end{aligned} \tag{4.25}$$

for some constants $c_0, \tilde{c}_0, b_9 > 0$.

Since $\epsilon_k \to 0$, if k is sufficiently large, then $\|\operatorname{Hess} f(x_j)^{-1}\| \|(\operatorname{Hess} f(x_j) - \mathcal{B}_j)s_j\| \le \frac{1}{2}\|s_j\|$. By (4.9), (4.17), (4.18), and Lemma 42, it follows that

$$\begin{aligned}
\|s_j\| &\le \|\operatorname{Hess} f(x_j)^{-1}\| \|(\operatorname{Hess} f(x_j) - \mathcal{B}_j)s_j\| + b_0\|g_j\| + b_0\|\delta_j\| \\
&\le \frac{1}{2}\|s_j\| + b_0 \lambda_2 e_j + b_0 \lambda_2^{1+\theta} e_j^{1+\theta}.
\end{aligned}$$

Therefore, $\|s_j\| \le 4 b_0 \lambda_2 e_j$ (note that e_j is sufficiently small such that $\lambda_2 e_j < 1$), which together with Lemma 46 and (4.25) implies that

$$
\begin{aligned}
\operatorname{dist}(x_{j+1}, x^*) &\leq b_7 \frac{\|(\mathcal{B}_j - \operatorname{Hess} f(x_j))s_j\|}{\|s_j\|}\|s_j\| + b_8 e_j^{1+\min\{\theta,1\}} \\
&\leq b_7 b_9 \epsilon_k^{1/d}\|s_j\| + b_8 e_j^{1+\min\{\theta,1\}} \\
&\leq 4 b_0 b_7 b_9 \lambda_2 \epsilon_k^{1/d} e_j + b_8 e_j^{1+\min\{\theta,1\}} \\
&= \left(4 b_0 b_7 b_9 \lambda_2 \epsilon_k^{1/d} + b_8 e_j^{\min\{\theta,1\}}\right) e_j .
\end{aligned}
$$

Since (4.24) holds for $N < k+1 \leq j \leq k+d$, we have

$$
\begin{aligned}
\lambda_3 e_{k+d+1}^2 &\leq f(x_{k+d+1}) - f(x^*) \\
&\leq f(x_{j+1}) - f(x^*) \\
&\leq \lambda_4 \operatorname{dist}^2(x_{j+1}, x^*) \\
&\leq \lambda_4 (4 b_0 b_7 b_9 \lambda_2 \epsilon_k^{1/d} + b_8 e_j^{\min\{\theta,1\}})^2 e_j^2 \\
&\leq \frac{\lambda_4}{\lambda_3}\left(4 b_0 b_7 b_9 \lambda_2 \epsilon_k^{1/d} + b_8 e_j^{\min\{\theta,1\}})^2 (f(x_j) - f(x^*)\right) \\
&\leq \frac{\lambda_4}{\lambda_3}\left(4 b_0 b_7 b_9 \lambda_2 \epsilon_k^{1/d} + b_8 e_j^{\min\{\theta,1\}}\right)^2 \left(f(x_k) - f(x^*)\right) \\
&\leq \frac{\lambda_4^2}{\lambda_3}\left(4 b_0 b_7 b_9 \lambda_2 \epsilon_k^{1/d} + b_8 e_j^{\min\{\theta,1\}}\right)^2 e_k^2 \\
&\leq \frac{\lambda_4^2}{\lambda_3}\left(4 b_0 b_7 b_9 \lambda_2 \epsilon_k^{1/d} + b_8 \epsilon_k^{\min\{\theta,1\}}\right)^2 e_k^2 .
\end{aligned}
$$

As k goes to ∞, ϵ_k tends to 0 , we obtain

$$
\frac{e_{k+d+1}}{e_k} \to 0,
$$

which proves our assertion. □

5 Numerical results

In this section, we demonstrate the effectiveness of our algorithm on some test problems. All of our tests are carried out in MATLAB R2014a on a Thinkpad notebook Intel Core i5 with 2.53GHz CPU and 4.00GB RAM .

In our subspace limited SR1 algorithm, τ_1 and τ_2 are set to be 0.1 and 0.75 respectively. The initial and maximum radius Δ is 1.0 and 2.0. We set c_1, c_2 to be 0.25, 2 respectively.

Since the convergence speed may slow down as the iterates approach a stationary point, it is critical to detect this and stop the algorithm properly. In addition, it is tricky to correctly predict whether an algorithm is temporarily or permanently trapped in a region when its convergence speed has reduced. Hence, it is usually beneficial to have flexible termination rules. In our implementation, in addition to checking the norm

Table 1 Comparison between LRTR-SR1 and SLRTR-SR1 over 10 tests

		n = 100, p = 3		n = 100, p = 5		n = 100, p = 10	
		LRTR-SR1	SLRTR-SR1	LRTR-SR1	SLRTR-SR1	LRTR-SR1	SLRTR-SR1
	cpu	0.11	0.10	0.21	0.19	0.34	0.28
m = 1	iter	110	126	213	220	296	296
	rad	7.52E−06	3.27E−06	9.18E−06	6.68E−06	9.25E−06	4.06E−06
	cpu	0.22	0.14	0.61	0.28	0.96	0.46
m = 2	iter	204	124	500	237	675	315
	rad	4.19E−06	3.32E−06	5.17E−06	4.94E−06	9.33E−06	4.61E−06

of the gradient $\|\text{grad} f(x)\| \leq \epsilon_g$, we also compute the relative changes of objective function values of the two consecutive iterates and terminate it as soon as

$$\frac{f(x_k) - f(x_{k+1})}{|f(x_k)| + 1} \leq \epsilon_f.$$

The default values of ϵ_f and ϵ_g are 10^{-8} and 10^{-7}. The max iterations is 1000.

We conduct the numerical experiment on two test problems: Brockett cost function and low rank completion.

5.1 Brockett cost function minimization

Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and $D = \text{diag}(\mu_1, \mu_2, \dots, \mu_p) \in \mathbb{R}^{p\times p}$ with $0 < \mu_1 < \mu_2 < \cdots < \mu_p$, where $n > p$, the Brockett cost function minimization problem can be formulated as

$$\begin{aligned} \min_{X\in\mathbb{R}^{n\times p}} \quad & \text{trace}(X^T AXD) \\ \text{s.t.} \quad & X^T X = I_p. \end{aligned} \tag{5.1}$$

We form a few randomly generated dense Wishart matrices assembled as $A = \bar{A}\bar{A}^T$, where $\bar{A} \in \mathbb{R}^{n\times n}$ is a matrix whose elements are sampled from the standard Gaussian distribution. In our numerical experiment, we choose $D = \text{diag}(1, 2, \dots, p)$ for convenience. The initial iterate X_0 is given by applying Matlab's function *orth* to a matrix whose elements are drawn from the standard normal distribution using Matlab's function *randn*.

We conduct our numberical testing on the problem above and compare the Riemannian subspace limited SR1 method (SLRTR-SR1 for short) with the Riemannian trust region limited SR1 method (LRTR-SR1 for short, see [12]) for the case $m = 1, 2$. We record the average numerical performance and list them in Table 1, in which 'iter' represents the iteration number, 'cpu' represents the required time and 'rad' represents the ratio of the norm of final gradient and the norm of initial gradient.

As shown in Table 1, both SLRTR-SR1 and LRTR-SR1 require fewer iterations and less computational time for $m = 2$ than $m = 1$. The computational cost will increase

Table 2 Numerical results of LRTR-BFGS, LRTR-SR1and SLRTR-SR1 for problem (5.1)

Parameter	LRTR-BFGS			LRTR-SR1			SLRTR-SR1		
	cpu	iter	rad	cpu	iter	rad	cpu	iter	rad
n = 100, various p (I)									
p = 5	0.50	193	3.14E−06	0.25	185	8.33E−06	0.19	171	4.96E−06
p = 10	0.70	299	7.51E−06	0.39	302	1.42E−05	0.29	267	4.79E−06
p = 15	1.67	582	7.05E−06	0.84	571	4.15E−06	0.59	465	4.21E−06
p = 5, various n (II)									
n = 300	0.76	261	6.52E−06	0.47	270	1.39E−05	0.36	243	3.37E−06
n = 500	1.22	288	4.64E−06	0.87	289	1.08E−05	0.66	257	3.87E−06
n = 1000	4.20	384	6.34E−06	3.51	404	6.49E−06	2.71	351	4.31E−06
n = 100, p = 20 (III)									
cond(A) $\approx 10^5$	1.49	580	4.28E−06	0.86	618	1.23E−05	0.63	539	3.07E−06
cond(A) $\approx 10^6$	1.53	602	6.36E−06	0.87	641	1.26E−05	0.64	545	3.40E−06
cond(A) $\approx 10^7$	1.60	625	5.06E−06	0.93	668	8.06E−06	0.64	550	3.96E−06

Table 3 Numerical results of RCG and SLRTR-SR1 for problem (5.2)

n, k	RCG				SLRTR-SR1			
	cpu	iter	obj	ng	cpu	iter	obj	ng
m = 100, n = 500, various rank k (I)								
k = 5	1.12	64	1.30E−11	8.89E−07	0.77	73	5.10E−12	5.56E−07
k = 10	1.01	47	3.77E−12	7.95E−07	0.65	49	6.56E−13	5.19E−07
k = 15	0.77	33	6.76E−13	6.27E−07	0.60	36	1.70E−13	4.00E−07
k = 20	0.60	26	7.90E−13	7.72E−07	0.44	25	1.41E−13	3.93E−07
m = 200, k = 10, various size n (II)								
n = 500	1.83	52	4.95E−12	7.75E−07	1.20	53	3.11E−12	6.79E−07
n = 1000	3.84	60	5.90E−12	6.33E−07	2.63	64	2.69E−12	4.89E−07
n = 1500	5.70	60	7.96E−12	8.56E−07	4.22	70	4.53E−12	5.59E−07
n = 2000	7.41	63	9.05E−12	9.07E−07	5.58	74	4.41E−12	5.66E−07

very fast as m increases, and so the two methods will lose their appeal. We will not present numerical results for the larger m. Overall, SLRTR-SR1 is competitive with LRTR-SR1 for this moderately sized problem.

The Riemannian trust region limited BFGS method is abbreviated as LRTR-BFGS (see [4]). Table 2 contains the results with various n, p and cond(A) for the LRTR-BFGS, LRTR-SR1 and SLRTR-SR1 (m = 1).

From Table 2 (I), for a fixed n, we can see that SLRTR-SR1 requires fewer iterations and less computational time to achieve a quite accurate solution than others. The advantage of the SLRTR-SR1 is more obvious especially when p grows larger, which indicates the efficiency of the subspace method. For a fixed p, Table 2 (II) show that the SLRTR-SR1 algorithm is competitive for large scale problems. For $n = 100$, $p = 20$, we investigate the influence of the condition number of the random matrix A on the algorithm in Table 2 (III). It is clear that the cpu time and iterations of our algorithm keep stable as the condition numbers grow. Overall, our algorithm is efficient and stable in most cases, even for ill-conditioned problems.

5.2 Low rank completion problem

Let $A \in \mathbb{R}^{m\times n}$ be an $m \times n$ matrix that is only known on a subset Ω of the complete set of entries $\{1, \ldots, m\} \times \{1, \ldots, n\}$. The low rank matrix completion is based on a direction optimization over the set of all fixed-rank matrices. By prescribing the rank, say k, the low rank completion problem in [17] is

$$\begin{aligned} \min_{X} \quad & \frac{1}{2}\|P_\Omega(X - A)\|_F^2 \\ \text{s.t.} \quad & X \in \mathcal{M}_k := \{X \in \mathbb{R}^{m\times n} : \operatorname{rank}(X) = k\}. \end{aligned} \tag{5.2}$$

in which $P_\Omega \in \mathbb{R}^{m\times n}$ denotes projection onto Ω. And we assume $m \leq n$.

We compared the SLRTR-SR1 with the Riemmanian conjugate gradient method in [7,17], abbreviated as 'RCG'. We also study the influence of the size and the rank for the methods. In Table 3, we report on the mean cpu time(cpu), the number of iterations(iter), the objective function value and the norm of the gradient(ng). For a fixed size of the matrices, it is clear that from Table 3 (I) our algorithm performs rather better than RCG especially on the run time, approximation, computational accuracy for small rank. For a fixed rank k, Table 3 (II) indicates that the iteration number of RCG seems to stop growing as the size n grows. However, although SLRTR-SR1 needs more iterations , it is faster and achieves lower objective function value than RCG in most cases.

Acknowledgments The work of Wei Hong Yang was supported by the National Natural Science Foundation of China No. 11371102 and NSFC Key Project 91330201.

References

1. Absil, P.-A., Baker, C.G., Gallivan, K.A.: A truncated CG style method for symmetric generalized eigenvalue problems. J. Comput. Appl. Math. **189**, 274–285 (2006)
2. Absil, P.-A., Baker, C.G., Gallivan, K.A.: Trust-region methods on Riemannian manifolds. Found. Comput. Math. **7**, 303–330 (2007)
3. Absil, P.-A., Baker, C.G., Gallivan, K.A.: Accelerated line-search and trust-region methods. SIAM J. Numer. Anal. **47**, 997–1018 (2009)
4. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
5. Baker, C.G.: Riemannian Manifold Trust-region Methods with Applications to Eigen problems. PhD thesis, School of Computational Science, Florida State University (2008)
6. Baker, C.G., Absil, P.-A., Gallivan, K.A.: An implicit trust-region method on Riemannian manifolds. IMA J. Numer. Anal. **28**, 665–689 (2008)
7. Boumal, N., Mishra, B., Absil, P.-A., Sepulchre, R.: Manopt, a Matlab toolbox for optimization on manifolds. J. Mach. Learn. Res. **15**, 1455–1459 (2014)
8. Byrd, R.H., Nocedal, J., Schnabel, R.B.: Representations of quasi-Newton matrices and their use in limited-memory methods. Math. Program. **63**, 129–156 (1994)

9. Byrd, R.H., Schnabel, R.B., Schultz, G.A.: Approximate solution of the trust regions problem by minimization over two-dimensional subspaces. Math. Program. **40**, 247–263 (1988)
10. Erway, J.B., Gill, P.E.: A subspace minimization method for the trust-region step. SIAM J. Optim. **20**, 1439–1461 (2009)
11. Gill, P.E., Leonard, M.W.: Reduced-Hessian quasi-Newton methods for unconstrained optimization. SIAM J. Optim. **12**, 209–237 (2001)
12. Huang, W.: Optimization Algorithms on Riemannian Manifolds with Applications. PhD thesis, Florida State University (2013)
13. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer Series in Operations Research and Financial Engineering. Springer, New York (2006)
14. Qi, C.H.: Numerical Optimization Methods on Riemannian Manifolds. PhD thesis, Florida State University (2011)
15. Ring, W., Wirth, B.: Optimization methods on Riemannian manifolds and their application to shape space. SIAM J. Optim. **22**, 596–627 (2012)
16. Sato, H., Iwai, T.: Optimization algorithms on the Grassmann manifold with application to matrix eigenvalue problems. Jpn. J. Ind. Appl. Math. **31**, 355–400 (2014)
17. Vandereycken, B.: Low-rank matrix completion by Riemannian optimization. SIAM J. Optim. **23**, 1214–1236 (2013)
18. Wang, Z., Wen, Z., Yuan, Y.: A subspace trust region method for large scale unconstrained optimization. In: Yuan, Y. (ed.) Numerical Linear Algebra and Optimization, pp. 265–274. Science Press, Beijing (2004)
19. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Math. Program. **142**, 397–434 (2013)
20. Wang, Z., Yuan, Y.: A subspace implementation of quasi-Newton trust region methods for unconstrained optimization. Numer. Math. **104**, 241–269 (2006)
21. Yuan, Y., Stoer, J.: A subspace study on conjugate gradient algorithms. Z. Angew. Math. Mech. **75**, 69–77 (1995)