Low-rank retractions: a survey and new results*

P.-A. Absil† I. V. Oseledets‡

October 31, 2013

Abstract

Retractions are a prevalent tool in Riemannian optimization that provides a way to smoothly select a curve on a manifold with given initial position and velocity. We review and propose several retractions on the manifold $\mathcal{M}_r$ of rank-r $m \times n$ matrices. With the exception of the exponential retraction (for the embedded geometry), which is clearly the least efficient choice, the retractions considered do not differ much in terms of run time and flop count. However, considerable differences are observed according to properties such as domain of definition, boundedness, first/second-order property, and symmetry.

Key words: Low-rank manifold; fixed-rank manifold; low-rank optimization; retraction; geodesic; quasi-geodesic; projective retraction; orthographic retraction; Lie-Trotter splitting

1 Introduction

We consider the general *low-rank optimization* problem of minimizing a real-valued function on a set of matrices of fixed rank:

$$\min_{X \in \mathcal{M}_r} f(X), \tag{1}$$

where

$$\mathcal{M}_r = \{X \in \mathbb{R}^{m\times n} : \mathrm{rank}(X) = r\} \tag{2}$$

is the set of $m \times n$ matrices of rank r and m, n, and $r < \min(m, n)$ are positive integers. Applications of (1) appear in particular in learning problems, where the low-rank constraint is inherent to the model or introduced to reduce memory usage and computation time; see the list of applications in the introduction of [MMBS13].

Several techniques have been proposed to address (1)—or more specific instances thereof—by exploiting the fact that $\mathcal{M}_r$ is a submanifold of the Euclidean space $\mathbb{R}^{m\times n}$; see, e.g., [MMBS11, SWC13, AAM13, MMBS13, Van13]. Most of these techniques choose a descent direction $\dot{X}$ for f

*This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. This work was financially supported by the Belgian FRFC (Fonds de la Recherche Fondamentale Collective).

†Department of Mathematical Engineering, ICTEAM Institute, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium (http://sites.uclouvain.be/absil/).

‡Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street 8, Moscow, Russia (ivan.oseledets@gmail.com).

in the tangent space to $\mathcal{M}_r$ at the current iterate $X \in \mathcal{M}_r$ and then compute the next iterate by performing a line search along a curve γ on $\mathcal{M}_r$ satisfying $\gamma(0) = X$ and $\frac{\mathrm{d}}{\mathrm{d}t}\gamma(t)\big|_{t=0} = \dot{X}$. The curve γ is conveniently chosen as $\gamma(t) = R(X, t\dot{X})$ where R is a *retraction* on $\mathcal{M}_r$. Retractions on manifolds, a concept due to Shub [Shu86] that we recall in Section 2.3, have received much attention lately in the context of Riemannian optimization; see, e.g., [ADM$^+$02, ABG07, AMS08, Van13, AM12, RW12, SWC13, BMAS13, KSV13].

The purpose of this paper is to review several retractions on $\mathcal{M}_r$ and propose new ones. In particular, we introduce the *Lie–Trotter retraction*, which directly follows from the first-order splitting method described in [LO13, §3.2]. An extended version of the "KSL" flavor of this retraction is known to have an exactness property [LO13, Theorem 4.1], from which we deduce that the Lie–Trotter retraction is a second-order retraction, i.e., the second derivative of $t \mapsto R(X, t\dot{X})$ at $t = 0$ belongs to the normal space to $\mathcal{M}_r$ at X.

The paper is organized as follows. After Section 2 giving the necessary background and preliminaries, Section 3 presents the various retractions, discussing their implementation and their computational cost. Numerical experiments comparing the retractions are conducted in Section 4. Conclusions are drawn in Section 5.

2 Background and preliminaries

This section recalls fundamental notions pertaining to the low-rank manifolds, flop counts, and retractions on manifolds.

2.1 The low-rank manifold

Central in this paper is the *low-rank manifold* $\mathcal{M}_r$ (2). This subsection gives background on the geometry of $\mathcal{M}_r$, with an emphasis on the representation of its elements and tangent vectors.

We first introduce some notation. Let

$$\mathrm{St}(r, m) = \{X \in \mathbb{R}^{m\times r} : X^\top X = I_r\}$$

denote the *(compact) Stiefel manifold* of orthonormal $m \times r$ matrices,

$$\mathbb{R}_*^{m\times r} = \{X \in \mathbb{R}^{m\times r} : \mathrm{rank}(X) = r\}$$

denote the *noncompact Stiefel manifold* of full column rank $m \times r$ matrices,

$$\mathrm{GL}(r) = \{X \in \mathbb{R}^{r\times r} : \mathrm{rank}(X) = r\}$$

denote the *general linear group* of order r, i.e., the set of all $r \times r$ invertible matrices, and

$$\mathrm{O}(r) = \{X \in \mathbb{R}^{r\times r} : X^\top X = I_r\}$$

denote the *orthogonal group* of order r, i.e., the set of all $r \times r$ orthogonal matrices.

The set $\mathcal{M}_r$ is known to be a submanifold of dimension $(m+n-r)r$ embedded in the Euclidean space $\mathbb{R}^{m\times n}$ [Lee03, Example 8.14]. The low-rank optimization problem (1) is thus in the field of play of Riemannian optimization; see, e.g., [AMS08].

In practice, we prefer not to store an $X \in \mathcal{M}_r$ as an $m \times n$ matrix; it requires storing mn numbers, which is much larger than the manifold dimension $(m+n-r)r$ in the frequent situation where $r \ll \min(m,n)$. Instead, we represent $X \in \mathcal{M}_r$ as

$$X = MN^\top \text{ with } (M,N) \in \mathcal{N}_1 := \mathbb{R}_*^{m\times r} \times \mathbb{R}_*^{n\times r}, \text{ or} \tag{3}$$

$$X = MN^\top \text{ with } (M,N) \in \mathcal{N}_2 := \mathrm{St}(r,m) \times \mathbb{R}_*^{n\times r}, \text{ or} \tag{4}$$

$$X = USV^\top \text{ with } (U,S,V) \in \mathcal{N} := \mathrm{St}(r,m) \times \mathrm{GL}(r) \times \mathrm{St}(r,n). \tag{5}$$

Several other representations exist, see [MMBS13, §3], but in this paper we will only make use of the three representations above, with an emphasis on (5). Note that the mappings

$$\pi_i : \mathcal{N}_i \to \mathcal{M}_r : (M,N) \mapsto MN^\top,$$

$i = 1, 2$, and

$$\pi : \mathcal{N} \to \mathcal{M}_r : (U,S,V) \mapsto USV^\top \tag{6}$$

are surjective (i.e., every $X \in \mathcal{M}_r$ is represented) but not injective: the equivalence classes of representations are

$$\begin{aligned}
\pi_1^{-1}(\pi_1(M,N)) &= \{(MR, NR^{-T}) : R \in \mathrm{GL}(r)\} \\
\pi_2^{-1}(\pi_2(M,N)) &= \{(MQ, NQ) : Q \in \mathrm{O}(r)\} \\
\pi^{-1}(\pi(U,S,V)) &= \{(UQ_U, Q_U^\top S Q_V, VQ_V) : Q_U, Q_V \in \mathrm{O}(r)\}.
\end{aligned} \tag{7}$$

We also point out, as we will allude to this fact later on, that each of the three "π" mappings is a submersion, i.e., its differential is surjective at every point; this is shown in detail in [AAM13, §2] for the case of π_1, and the two other cases can be treated similarly. This provides us with three different expressions of $\mathcal{M}_r$ as a quotient manifold. The one that concerns us most is

$$\mathcal{M}_r \simeq (\mathrm{St}(r,m) \times \mathrm{GL}(r) \times \mathrm{St}(r,n))/(\mathrm{O}(r) \times \mathrm{O}(r)) \tag{8}$$

with quotient map (6) whose fibers are given by (7).

The set of all tangent vectors to $\mathcal{M}_r$ at $X = USV^\top$ (5) is termed the *tangent space* to $\mathcal{M}_r$ at X and denoted by $\mathrm{T}_X\mathcal{M}_r$. The concept of tangent vector to an abstract manifold can be found, e.g., in [Boo03] or [AMS08]. Since $\mathcal{M}_r$ is a submanifold of $\mathbb{R}^{m\times n}$, the tangent space $\mathrm{T}_X\mathcal{M}_r$ is simply identified with $\{\gamma'(0) : \gamma \text{ smooth curve on } \mathcal{M}_r \text{ with } \gamma(0) = X\}$. Depending of whether we want to recall the foot X in the notation, we write $(X, \dot{X}) \in \mathrm{T}_X\mathcal{M}_r$ or $\dot{X} \in \mathrm{T}_X\mathcal{M}_r$.

The projection $\mathcal{P}_X Z$ of $Z \in \mathbb{R}^{m\times n}$ onto the tangent space $\mathrm{T}_X\mathcal{M}_r$ is given by

$$\mathcal{P}_X Z = ZVV^\top + UU^\top Z - UU^\top ZVV^\top, \tag{9}$$

see [KL07, Lemma 4.1]. Hence every tangent vector $\dot{X}$ to $\mathcal{M}_r$ at $X \in \mathcal{M}_r$ can be written in the form

$$\dot{X} = ZVV^\top + UU^\top Z - UU^\top ZVV^\top. \tag{10}$$

However, the choice of Z to represent $\dot{X}$ is not unique, and moreover $Z \in \mathbb{R}^{m\times n}$ contains again mn numbers, to be compared with the dimension $(m+n-r)r$ of the vector space $\mathrm{T}_X\mathcal{M}_r$. These drawbacks are remedied next.

Once a decomposition (5) is chosen for $X \in \mathcal{M}_r$, a unique representation $(\dot{U}, \dot{S}, \dot{V})$ of any $\dot{X} \in \mathrm{T}_X\mathcal{M}_r$ can be chosen such that

$$\dot{X} = U\dot{S}V^\top + \dot{U}SV^\top + US\dot{V}^\top, \tag{11a}$$

$$U^\top\dot{U} = 0, \quad V^\top\dot{V} = 0. \tag{11b}$$

This follows from [KL07, §2.1], or alternatively from the machinery of quotient manifolds by showing that $(U, S, V) \mapsto \{(\dot{U}, \dot{S}, \dot{V}) : U^\top\dot{U} = 0,\ V^\top\dot{V} = 0\}$ is a horizontal distribution for the quotient (8). If $\dot{X} = \mathcal{P}_X Z$, and in particular if $\dot{X} = Z$, the decomposition (11) is given by

$$\begin{aligned}
\dot{S} &= U^\top ZV, \\
\dot{U} &= (ZV - U\dot{S})S^{-1} = (I - UU^\top)ZVS^{-1}, \\
\dot{V} &= (Z^\top U - V\dot{S}^\top)S^{-T} = (I - VV^\top)Z^\top US^{-T}.
\end{aligned}$$

In order to get rid of the inverse of S that appears in the above formulas, we can set $U_p := \dot{U}S$ and $V_p = \dot{V}S$, which yields the unique representation considered in [Van13, §2.1]:

$$\dot{X} = U\dot{S}V^\top + U_pV^\top + UV_p^\top, \tag{12a}$$

$$U^\top U_p = 0,\ V^\top V_p = 0. \tag{12b}$$

We will favor this representation. If $\dot{X} = \mathcal{P}_X Z$, and in particular if $\dot{X} = Z$, the decomposition (12) is given by

$$\begin{aligned}
\dot{S} &= U^\top ZV, \\
U_p &= ZV - U\dot{S} = (I - UU^\top)ZV, \\
V_p &= Z^\top U - V\dot{S}^\top = (I - VV^\top)Z^\top U.
\end{aligned}$$

Finally, the *normal space* at X is the orthogonal complement to $\mathrm{T}_X\mathcal{M}_r$, in the sense of the classical Frobenius inner product in the embedding space $\mathbb{R}^{m\times n}$. Since the tangent space is given by

$$\mathrm{T}_X\mathcal{M}_r = \{U\dot{S}V^\top + U_pV^\top + UV_p^\top : \dot{S} \in \mathbb{R}^{r\times r},\ U_p \in \mathbb{R}^{m\times r},\ U^\top U_p = 0,\ V_p \in \mathbb{R}^{n\times k},\ V^\top V_p = 0\}$$

one finds that the normal space is

$$T_X^\perp\mathcal{M}_r = \{Z \in \mathbb{R}^{m\times n} : U^\top Z = 0 \text{ and } ZV = 0\}. \tag{13}$$

2.2 Flop counts

We will use flop counts as a way to compare the computational cost of various operations. When counting flops, we follow the conventions in [GV96, §1.2.4], and we always assume, unless otherwise stated, that $m < n$, as this has an impact on the way certain computations are carried out. For example, $U^\top ZV$ costs $2mnr + 2mr^2$ flops when performed as $U^\top(ZV)$ and $2mnr + 2nr^2$ flops when performed as $(U^\top Z)V$; the assumption that $m < n$ will thus lead us to prefer the $U^\top(ZV)$ order, unless $U^\top Z$ is needed in later operations and ZV not, in which case the $(U^\top Z)V$ order is preferable. These are subtle considerations that need to be taken into account when looking for the most flop-efficient implementation. This calls for two caveats. The first one is that we have made a careful but not systematic attempt to find the most flop-efficient implementation of the various retractions studied in this paper.

The second caveat, much in line with the last paragraph of [GV96, §1.2.4], is that the flop count does not directly translate into time efficiency, as it ignores various overheads—such as memory traffic—associated with program execution. For example, if A and B are $m \times r$ and C is $r \times r$ with $m = 10^4$ and $r = 10^2$, then computing `A'*B` in Matlab can be significantly slower than computing `A*C`, even though both operations have the same the flop count of $2mr^2$. Moreover, even though it can also be performed in $2mr^2$ flops with modified Gram-Schmidt [GV96, §5.2.8], the economy-size QR decomposition of A in Matlab is slower than the product `A*C`, even much so on older versions of Matlab. In order to avoid relying on Matlab's QR implementation, we perform orthonormalizations using the polar decomposition $A = QP$ with $P = (A^\top A)^{1/2}$ and $Q = A/P$, for a dominant flop count of $4mr^2$ when A is of size $m \times r$ with $r \ll m$.

These remarks indicate that the flop counts and the timing comparisons given in this paper, while informative, must be taken with a grain of salt. With these provisions in mind, we obtain that computing $\dot{X} = \mathcal{P}_X Z$ (9) in the $(U_p, \dot{S}, V_p)$ representation (12), with X given in the (U, S, V) representation (5), requires a flop count of $2mnr[ZV] + 2mr^2[U^\top ZV] + 2mr^2[U\dot{S}] + 2mnr[Z^\top U] + 2nr^2[V\dot{S}^\top]$, hence $4mnr + 4mr^2 + 2nr^2$.

Throughout the paper, we present dominant flop counts under the common assumption that the rank is very low, i.e., $r \ll m$. For $\mathcal{P}_X Z$, the dominant flop count is thus $4mnr$.

2.3 Retractions on manifolds

A *(first-order) retraction* [ADM+02] (or see [AMS08, §4.1]) on a manifold $\mathcal{M}$ is a smooth mapping R from the tangent bundle $T\mathcal{M}$ onto $\mathcal{M}$ such that

1. R is defined and smooth on a neighborhood of the zero section in $T\mathcal{M}$;
2. $R(X, 0) = X$ for all $X \in \mathcal{M}$;
3. $\frac{\mathrm{d}}{\mathrm{d}t} R(X, t\dot{X})\Big|_{t=0} = \dot{X}$ for all $X \in \mathcal{M}$ and $\dot{X} \in \mathrm{T}_X\mathcal{M}$.

When $\mathcal{M}$ is an embedded submanifold of a Euclidean space $\mathcal{E}$, which is the case of the low-rank manifold $\mathcal{M}_r$, we say that a retraction R is a *second-order retraction* if moreover $\frac{\mathrm{d}^2}{\mathrm{d}t^2} R(X, t\dot{X})\Big|_{t=0}$ belongs to the normal space at X to $\mathcal{M}$ in $\mathcal{E}$. This is one of the criteria along which the retractions described in Section 3 differ; see in particular the discussion in Section 4.2.

Retractions are useful in optimization algorithms for applying an update vector $\dot{X}$ to a current point X [AMS08, §4] or for "lifting" an objective function to the tangent space [AMS08, §7].

In the present paper, we also consider a related concept which we call *extended retraction*, defined as a mapping R from $\{\mathrm{T}_X\mathcal{E} : X \in \mathcal{M}\} \simeq \{(X, Z) : X \in \mathcal{M}, Z \in \mathcal{E}\}$ onto $\mathcal{M}$ such that $R(X, 0) = X$ and $\frac{\mathrm{d}}{\mathrm{d}t} R(X, tZ)\big|_{t=0} = \mathcal{P}_X Z$ for all $X \in \mathcal{M}$ and $Z \in \mathbb{R}^{m \times n}$, where $\mathcal{P}_X$ denotes the orthogonal projector onto the tangent space to $\mathcal{M}$ at X. (In the case of $\mathcal{M}_r$, $\mathcal{P}$ is given by (9).) This new concept will be exploited in Section 3.7.

3 Retractions on the low-rank manifold

In this central section, we present and analyze several retractions on the low-rank manifold $\mathcal{M}_r$. At the beginning of each subsection, we mention the related literature of which we are aware. In particular, the developments in Sections 3.5 and 3.9 are new to the best of our knowledge. The Lie–Trotter retractions of Sections 3.7 and 3.8 are arguably new as (extended) retractions but they directly follow from the material in [LO13, §3.2].

3.1 Projective retraction

The projective retraction is perhaps the retraction that most directly comes to mind on submanifolds embedded in a Euclidean space. It consists in defining $R(X, \dot{X})$ as the projection of $X + \dot{X}$ on the manifold. The projective retraction on the low-rank manifold $\mathcal{M}_r$ is described and analyzed in [AM12, §3.2], and an efficient implementation is presented in [Van13, §3].

The projective retraction is thus defined by

$$R(X, \dot{X}) = \underset{Y \in \mathcal{M}_r}{\arg\min} \|Y - (X + \dot{X})\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Let $\sigma_1(A), \ldots, \sigma_{\min(m,n)}(A)$ be the singular values of an $m \times n$ matrix A in decreasing order. It is known [AM12, Prop. 6] that whenever $\dot{X}$ is sufficiently small for $\|\dot{X}\| < \sigma_r(X)/2$ to hold, then $R(X, \dot{X})$ exists, is unique, and

$$R(X, \dot{X}) = \sum_{i=1}^{r} \sigma_i u_i v_i,$$

where $X + \dot{X} = \begin{bmatrix} u_1 & \ldots & u_{\min(m,n)} \end{bmatrix} \operatorname{diag}(\sigma_1, \ldots, \sigma_{\min(m,n)}) \begin{bmatrix} v_1 & \ldots & v_{\min m,n} \end{bmatrix}^\top$ is a singular value decomposition (SVD) with singular values in decreasing order.

If $\dot{X}$ is available in the form (12), then [Van13, §3] shows that $R(X, \dot{X})$ can be computed efficiently as follows. First perform orthonormalizations $U_p = Q_u S_u$ and $V_p = Q_v S_v$. Observe that

$$X + \dot{X} = \begin{bmatrix} U & Q_u \end{bmatrix} \begin{bmatrix} S + \dot{S} & S_u \\ S_v^\top & 0 \end{bmatrix} \begin{bmatrix} V^\top \\ Q_v^\top \end{bmatrix}.$$

Obtain (U_s, Σ_s, V_s) as an SVD (with decreasing singular values) of the small $2r$-by-$2r$ matrix $\begin{bmatrix} S + \dot{S} & S_u \\ S_v^\top & 0 \end{bmatrix}$. Then we have

$$R(X, \dot{X}) = U_+ S_+ V_+^\top,$$

where $U_+ = \begin{bmatrix} U & Q_u \end{bmatrix} U_s(:, 1:r)$, $V_+ = \begin{bmatrix} V & Q_v \end{bmatrix} V_s(:, 1:r)$, and $S_+ = \Sigma_s(1:r, 1:r)$.

It is shown in [AM12] that the projective retraction is a second-order retraction.

3.1.1 Matlab implementation details

In the numerical experiments of Section 4, this retraction is labeled `proj`. The orthonormalizations are obtained with the polar decomposition mentioned in Section 2.2. The SVD is computed with the `svd` function, which we found to be faster than getting the truncated SVD directly with `svds`.

3.1.2 Flop count

Unless otherwise stated, we assume throughout the paper that X and $\dot{X}$ are given in the form (5) and (12), and we consider the dominant flop count when $r \ll m \leq n$.

The dominant flop count to compute (U_+, S_+, V_+) is then $4mr^2[Q_u, S_u] + 4nr^2[Q_v, S_v] + O(r^3)[U_s, \Sigma_s, V_s] + 2m2r^2[U_+] + 2n2r^2[V_+]$, hence $8(m+n)r^2$. However, one should bear in mind that the $O(r^3)$ flop count of the SVD concerns an operation that cannot in general be performed with finitely many elementary arithmetic operations; it thus involves an iterative process and a stopping criterion, and this may have a significant impact on the computation time when r is not much smaller than m. This comment also applies to the matrix square root involved in the polar decomposition.

If a line search has to be performed, it is also informative to estimate the additional flop count required to compute $R(X, t\dot{X})$ for a new value of t. In the case of the projective retraction, this additional flop count is $O(r^3)[U_s, \Sigma_s, V_s] + 2m2r^2[U_+] + 2n2r^2[V_+]$, hence a dominant flop count of $4(m+n)r^2$.

3.1.3 Inverse retraction

Computing the inverse retraction is required in certain situations, e.g., the computation of the R-barycenter X of a collection of points $Y_1, \ldots, Y_N$, defined by $\sum_{i=1}^N R_X^{-1} Y_i = 0$, where $R_X^{-1} Y_i$ stands for the tangent vector $\dot{X}$ at X such that $R(X, \dot{X}) = Y_i$. Note however that, depending on the manifold and the retraction, the R-barycenter and the inverse retractions may not be uniquely defined.

We are not aware of a computationally tractable way of computing the inverse projective retraction. The retraction considered next, however, has a very simple inverse.

3.2 Orthographic retraction

The orthographic retraction on $\mathcal{M}_r$ is introduced in [AM12, §4.4], but computational aspects are not discussed therein. The concept can be found as far back as [Ros61, Lue72].

The orthographic retraction R on $\mathcal{M}_r$ is defined by setting $R(X, \dot{X})$ as the point nearest to $X + \dot{X}$ in

$$X + \dot{X} + T_X^{\perp}\mathcal{M}_r \;\cap\; \mathcal{M}_r. \tag{14}$$

This point is unique when $\dot{X}$ is sufficiently small. When X and $\dot{X}$ are represented as in (5) and (12), $R(X, \dot{X})$ can be expressed as follows:

$$\begin{aligned} R(X, \dot{X}) &= (U(S + \dot{S}) + U_p)(S + \dot{S})^{-1}((S + \dot{S})V^\top + V_p^\top) \\ &= U_+ S_+ V_+^\top, \end{aligned}$$

where $U(S + \dot{S}) + U_p =: U_+ S_U$ and $V(S^\top + \dot{S}^\top) + V_p =: V_+ S_V$ are orthonormalizations and $S_+ := S_U (S + \dot{S})^{-1} S_V^\top$.

By virtue of the analysis in [AM12], this is a second-order retraction.

3.2.1 Matlab implementation details

This retraction is labeled `orth`. In our Matlab implementation, the orthonormalizations are obtained with the polar decomposition, and we use `mldivide` in the computation of S_+.

3.2.2 Flop count

Under the standing assumptions stated in Section 3.1.2, the flop count is $2mr^2[U(S+\dot{S})+U_p] + 2nr^2[V(S^\top+\dot{S}^\top)+V_p] + 4mr^2[U_+, S_U] + 4nr^2[V_+, S_V] + O(r^3)[S_+]$. The dominant cost is thus $6(m+n)r^2$.

The dominant additional flop count to compute $R(X, t\dot{X})$ for a new value of t can be reduced to $2(m+n)r^2$ if adequate matrices are precomputed (namely, US, $U\dot{S}+U_p$, $U^\top U_p$, $U_p^\top U_p$, $VS^\top$, $V\dot{S}^\top + V_p$, $V^\top V_p$, and $V_p^\top V_p$).

3.2.3 Inverse retraction

The inverse orthographic retraction is simple:

$$R_X^{-1}Y = \mathcal{P}_X(Y-X) = YVV^\top + UU^\top Y - UU^\top YVV^\top - X,$$

where $\mathcal{P}$ is the projection (9). If $Y = U_Y S_Y V_Y^\top$, we have

$$R_X^{-1}Y = ((I-UU^\top)U_Y S_Y V_Y^\top V)V^\top + U(U^\top U_Y S_Y V_Y^\top (I-VV^\top)) + U(U^\top U_Y S_Y V_Y^\top V - S)V^\top.$$

This yields the form (12) for $R_X^{-1}Y$.

3.3 Quotient-based retraction: compact Stiefel approach

A retraction that one naturally obtains by viewing $\mathcal{M}_r$ as the quotient (8) is the following:

$$R(X, \dot{X}) = U_+ S_+ V_+^\top,$$

where

$$\begin{aligned} U_+ &= R_{\mathrm{St}}(U, \dot{U}) \\ S_+ &= S + \dot{S} \\ V_+ &= R_{\mathrm{St}}(V, \dot{V}), \end{aligned}$$

and R_{St} is a retraction on the corresponding Stiefel manifold. Since, as we have seen in (7), the decomposition (5) of X is not unique, we need to ensure that the outcome $U_+S_+V_+^\top$ does not depend on the choice of the decomposition. This invariance is seen to hold if and only if the retraction on Stiefel satisfies $R_{\mathrm{St}}(UQ_U, \dot{U}Q_U) = R_{\mathrm{St}}(U, \dot{U})Q_U$ for all $Q_U \in \mathrm{O}(r)$. One such retraction is the projective retraction on Stiefel [AM12, §3.3], advocated in [MS13, §3.3], which returns the orthonormal factor of the polar decomposition of $U + \dot{U}$.

3.3.1 Matlab implementation details

We use the projective retraction on Stiefel for R_{St}. The resulting retraction on $\mathcal{M}_r$ is labeled `StRSt_pj`.

3.3.2 Flop count

Assuming as usually that $\dot{X}$ is provided in the representation (12), the dominant flop count is $2mr^2[U + U_pS^{-1}] + 4mr^2[U_+] + 2nr^2[V + V_pS^{-T}] + 4nr^2[V_+]$, that is, $6(m+n)r^2$.

The dominant flop count to compute $R(X, t\dot{X})$ for a new t can be reduced to $2(m+n)r^2$ if adequate matrices are precomputed.

3.3.3 Inverse retraction

Assume that the projective retraction is used on Stiefel. Given $X = USV^\top$ and $Y = U_+S_+V_+^\top$ in $\mathcal{M}_r$, we seek $\dot{X} \in \mathrm{T}_X\mathcal{M}_r$ such that $R(X, \dot{X}) = Y$. Observe that $Y = (U_+Q_U)(Q_U^\top S_+Q_V)(V_+Q_V)^\top$ for all Q_U, Q_V orthogonal. We need $U_+Q_U = (U + \dot{U})P_U$ with P_U symmetric positive definite, and this yields $U_+^\top U =: Q_UP_U$ (polar decomposition) and $\dot{U} = U_+Q_UP_U^{-1} - U$. Likewise, we set $V_+^\top V =: Q_VP_V$ (polar decomposition) and $\dot{V} = V_+Q_VP_V^{-1} - V$. Finally, $\dot{S} = Q_U^\top S_+Q_V - S$, and we have

$$R^{-1}_{USV^\top}(\dot{X}) = U_+S_+V_+^\top,$$

where $\dot{X}$ is given by (11).

3.4 Quotient-based retraction: noncompact Stiefel approach

Yet another possibility is to define

$$\begin{aligned} R(X, \dot{X}) &= (U + \dot{U})(S + \dot{S})(V + \dot{V})^\top \\ &= (US + U_p)S^{-1}(S + \dot{S})S^{-1}(VS^\top + V_p)^\top \\ &= U_+S_+V_+^\top, \end{aligned}$$

where $US+U_p = U_+S_U$ and $VS^\top+V_p = V_+S_V$ are two orthonormalizations and $S_+ = S_US^{-1}(S+\dot{S})S^{-1}S_V^\top$. This is a retraction that one naturally obtains by viewing $\mathcal{M}_r$ as the quotient manifold $(\mathbb{R}_*^{m\times r} \times \mathrm{GL}(r) \times \mathbb{R}_*^{n\times r})/(\mathrm{GL}(r) \times \mathrm{GL}(r))$ and favoring representations where the first and third factors are orthonormal.

3.4.1 Matlab implementation details

This retraction is labeled `RRR`. We use the polar decomposition for the orthonormalizations.

3.4.2 Flop count

Assuming that the orthonormalizations are chosen as polar decompositions, the dominant flop count is $2mr^2[US + U_p] + 4mr^2[U_+, S_U] + 2nr^2[VS^\top + V_p] + 4nr^2[V_+, S_+]$, that is, $6(m+n)r^2$.

The dominant flop count to compute $R(X, t\dot{X})$ for a new t is reduced to $4(m+n)r^2$ if US and $VS^\top$ are precomputed, and even $2(m+n)r^2$ if $U^\top\dot{U}$, $\dot{U}^\top\dot{U}$, $V^\top\dot{V}$, and $\dot{V}^\top\dot{V}$ are also precomputed.

3.4.3 Inverse retraction

We seek $\dot{X}$ in the form (11) such that $R^{-1}_{USV^\top}(\dot{X}) = U_+S_+V_+^\top$. Observe that $U_+S_+V_+^\top = (U_+S_U)(S_U^{-1}S_+S_V^{-T})(V_+S_V)^\top$ for all S_U, S_V invertible. We thus require $U + \dot{U} = U_+S_U$, $S + \dot{S} = S_U^{-1}S_+S_V^{-T}$, and $V + \dot{V} = V_+S_V$. This yields $S_U = (U^\top U_+)^{-1}$, $\dot{U} = U_+S_U - U$, $S_V = (V^\top V_+)^{-1}$, $\dot{V} = V_+S_V - V$, $\dot{S} = S_U^{-1}S_+S_V^{-T} - S$.

3.5 Simple second-order retractions

The retractions proposed in this section are more conveniently derived using a two-factor approach. Without loss of generality, following the notation of [AAM13], we thus consider $X = MN^\top \in \mathcal{M}_r$ and $\dot{X} = MHN^\top + M_\perp KN^\top + MLN_\perp^\top \in \mathrm{T}_X\mathcal{M}_r$, where $M \in \mathbb{R}_*^{m\times r}$, $N \in \mathbb{R}_*^{n\times r}$, $H \in \mathbb{R}^{r\times r}$, $K \in \mathbb{R}^{(m-r)\times r}$, $L \in \mathbb{R}^{r\times(n-r)}$, the columns of $M_\perp$ form a basis of the orthogonal complement of the column space of M, and likewise for $N_\perp$ with N. Our goal is to derive an expression for $R(X, \dot{X})$ that makes R is a second-order retraction on $\mathcal{M}_r$.

We seek R in the form

$$R(X, \dot{X}) = \begin{bmatrix} M & M_\perp \end{bmatrix} \begin{bmatrix} A_0 + A_1 + A_2 \\ B_0 + B_1 + B_2 \end{bmatrix} \begin{bmatrix} C_0 + C_1 + C_2 & D_0 + D_1 + D_2 \end{bmatrix} \begin{bmatrix} N^\top \\ N_\perp^\top \end{bmatrix},$$

where terms indexed by j ($j = 0, 1, 2$) are jth order expressions of $\dot{X}$. The "0th order" condition on R (i.e., $R(X, 0) = X$) yields

$$A_0C_0 = I \quad (16a)$$
$$B_0C_0 = 0 \quad (16b)$$
$$A_0D_0 = 0 \quad (16c)$$
$$B_0D_0 = 0. \quad (16d)$$

The first-order condition on R (i.e., $\frac{\mathrm{d}}{\mathrm{d}t}R(X, t\dot{X})\Big|_{t=0} = \dot{X}$) yields

$$A_1C_0 + A_0C_1 = H \quad (17a)$$
$$B_0C_1 + B_1C_0 = K \quad (17b)$$
$$A_0D_1 + A_1D_0 = L \quad (17c)$$
$$B_0D_1 + B_1D_0 = 0. \quad (17d)$$

Finally, the second-order condition on R (i.e., $\frac{\mathrm{d}^2}{\mathrm{d}t^2}R(X, t\dot{X})\Big|_{t=0} \in T_X^\perp\mathcal{M}_r$) yields

$$A_0C_2 + A_1C_1 + A_2C_0 = 0 \quad (18a)$$
$$B_0C_2 + B_1C_1 + B_2C_0 = 0 \quad (18b)$$
$$A_0D_2 + A_1D_1 + A_2D_0 = 0 \quad (18c)$$
$$B_0D_2 + B_1D_1 + B_2D_0 \text{ arbitrary}. \quad (18d)$$

The above system of matrix equations is underdetermined. A simple solution is readily found to be

$$A_0 = C_0 = I,\ B_0 = D_0 = 0,$$
$$A_1 = H,\ C_1 = 0,\ B_1 = K,\ D_1 = L,$$
$$A_2 = 0,\ C_2 = 0,\ B_2 = 0,\ D_2 = -HL.$$

The resulting retraction is thus given by

$$R(X,\dot{X}) = \begin{bmatrix} M & M_p \end{bmatrix} \begin{bmatrix} I+H \\ I \end{bmatrix} \begin{bmatrix} I & I-H \end{bmatrix} \begin{bmatrix} N^\top \\ N_p^\top \end{bmatrix},$$

where we set $M_p = M_\perp K$ and $N_p = N_\perp L^\top$.

In the three-factor representation (5) and (12), this yields

$$R(X,\dot{X}) = U_+ S_+ V_+^\top \tag{19}$$

where $U_+S_U := U(S+\dot{S}) + U_p$ and $V_+S_V := V + V_p S^{-T}(I - \dot{S}^\top S^{-T})$ are orthonormalizations and $S_+ := S_U S_V^\top$. To see this, link the 2-factor and 3-factor representations by taking $M = U$ and $N = VS^\top$; the relation between the tangent vector representations is then seen to be given by $\dot{S} = HS$, $U_p = M_\perp KS$, $V_p = N_\perp L^\top$, and the retraction then writes

$$R(X,\dot{X}) = \begin{bmatrix} U & U_p S^{-1} \end{bmatrix} \begin{bmatrix} I+\dot{S}S^{-1} \\ I \end{bmatrix} \begin{bmatrix} S & I-\dot{S}S^{-1} \end{bmatrix} \begin{bmatrix} V^\top \\ V_p^\top \end{bmatrix}.$$

3.5.1 Matlab implementation details

This retraction is labeled `ez-2nd`. We use the polar decomposition for the orthonormalizations. We explicitly compute S^{-1} using `inv` as it appears twice.

3.5.2 Flop count

Assuming as usually that X as in (5) and $\dot{X}$ as in (12) are provided, the dominant flop count is $2mr^2$ $[U(S+\dot{S}) + U_p]$ + $2nr^2$ $[V + V_p S^{-T}(I - \dot{S}^\top S^{-T})]$ + $4mr^2$ $[U_+]$ + $4nr^2$ $[V_+]$, hence $6(m+n)r^2$.

The dominant flop count to compute $R(X, t\dot{X})$ for a new t can be reduced to $2(m+n)r^2$ with adequate precomputed matrices.

3.6 Simple second-order balanced retraction

In the underdetermined system of equations obtained in Section 3.5, if we moreover impose a better balancing between the left and right factors by further requiring that $A_1 = C_1$, then we are led to the retraction proposed in [SWC13, Lemma 4]. In the representation (5) and (12), it writes as follows:

$$R(X,\dot{X}) = U_+ S_+ V_+^\top, \tag{20}$$

where $U_+S_U := U(S + \frac{1}{2}\dot{S} - \frac{1}{8}\dot{S}S^{-1}\dot{S}) + U_p(I - \frac{1}{2}S^{-1}\dot{S})$ and $V_+S_V := V(S^\top + \frac{1}{2}\dot{S}^\top - \frac{1}{8}\dot{S}^\top S^{-T}\dot{S}^\top) + V_p(I - \frac{1}{2}S^{-T}\dot{S}^\top)$ are orthonormalizations and $S_+ := S_U S^{-1} S_V^\top$.

3.6.1 Matlab implementation details

This retraction is labeled `Shalit`. The same comments as those of Section 3.5 apply.

3.6.2 Flop count

The dominant flop count is the same as in Section 3.5.

3.7 Lie–Trotter extended retraction

Observe that the three terms in $\mathcal{P}_X Z$ (9) belong to $\mathrm{T}_X\mathcal{M}_R$. Following [LO13, §3.2], let us define the *KSL Lie–Trotter extended retraction* R on $\mathcal{M}_r$ by setting $R(X,Z)$ as follows for all $Z \in \mathrm{T}_X\mathbb{R}^{m\times n} \simeq \mathbb{R}^{m\times n}$, thus in particular for all $Z \in \mathrm{T}_X\mathcal{M}_r$:

1. Obtain U_1 and $\hat{S}_1$ from
$$U_1\hat{S}_1V = USV + ZVV^\top. \tag{21}$$
One gets U_1 and $\hat{S}_1$ by an orthonormalization $U_1\hat{S}_1 = US + ZV$ with U_1 orthonormal.

2. Obtain $\tilde{S}_0$ from
$$U_1\tilde{S}_0V^\top = U_1\hat{S}_1V^\top - U_1U_1^\top ZVV^\top. \tag{22}$$
One gets $\tilde{S}_0$ by $\tilde{S}_0 = \hat{S}_1 - U_1^\top ZV$.

3. Obtain V_1 and S_1 from
$$U_1S_1V_1^\top = U_1\tilde{S}_0V^\top + U_1U_1^\top Z. \tag{23}$$
One gets V_1 and S_1 by an orthonormalization $V_1S_1^\top = V\tilde{S}_0^\top + Z^\top U_1$.

Finally, set
$$R(X,Z) = U_1S_1V_1^\top. \tag{24}$$

3.7.1 Analysis

We now need to show that the above procedure indeed defines an extended retraction on $\mathcal{M}_r$.

First, it can be shown that $R(X,Z)$ is well defined, i.e., the outcome (24) does not depend on the choice of the representation (5) of X nor on the orthonormalizations that yield U_1 and V_1. To see this, consider the above procedure carried out with two representations $X = USV^\top = \underline{USV}^\top$, and use the underline notation to denote the results obtained with the second representation. One has, for some $Q_U, Q_V \in \mathrm{O}(r)$, $\underline{U} = UQ_U$, $\underline{V} = VQ_V$, and $\underline{S} = Q_U^\top SQ_V$. We then find that, for some $Q_{U_1}, Q_{V_1} \in \mathrm{O}(r)$, we have $\underline{U}_1 = UQ_{U_1}$, $\underline{\hat{S}}_1 = U_{U_1}^\top\hat{S}_1Q_V$, $\underline{\tilde{S}}_0 = Q_{U_1}^\top\tilde{S}_0Q_V$, $\underline{V}_1 = V_1Q_{V_1}$, and $\underline{\tilde{S}}_1 = Q_{U_1}^\top S_1Q_{V_1}$, which yields that $U_1S_1V_1^\top = \underline{U}_1\underline{S}_1\underline{V}_1^\top$. For example, the first step yields $U_1\hat{S}_1 = US + ZV$ and $\underline{U}_1\underline{\hat{S}}_1 = \underline{US} + Z\underline{V}$, which yields the above relations $\underline{U}_1 = UQ_{U_1}$, $\underline{\hat{S}}_1 = U_{U_1}^\top\hat{S}_1Q_V$.

Second, the mapping R is smooth. This is readily seen by choosing the polar decomposition for the orthonormalizations and noting that the polar factors are smooth functions of their product.

Third, it is readily checked that the zeroth-order property holds: $R(X,0) = X$ for all $X \in \mathcal{M}_r$.

The forth and final point is to show the first order property: $\frac{\mathrm{d}}{\mathrm{d}t}R(X,tZ)\big|_{t=0} = \mathcal{P}_X Z$ for all $X \in \mathcal{M}$ and $Z \in \mathbb{R}^{m\times n}$. To this end, first observe that if $U(t)S(t) = A(t)$ is a time-varying polar decomposition, then $U' = (I - UU^\top)A'S^{-1} + U\mathrm{skew}(U^\top A')S^{-1}$ and $S' = \mathrm{sym}(U^\top A')$, where $\mathrm{skew}(A) = \frac{1}{2}(A - A^\top)$ and $\mathrm{sym}(A) = \frac{1}{2}(A + A^\top)$ denote the skew-symmetric and symmetric components of A. This can be deduced using the product rule and the expression $\mathrm{T}_U\mathrm{St}(m,r) = \{U\Omega + U_\perp K : \Omega = -\Omega^\top \in \mathbb{R}^{r\times r}, K \in \mathbb{R}^{(m-r)\times r}\}$ where $U_\perp$ is such that $\begin{bmatrix} U & U_\perp \end{bmatrix} \in \mathrm{O}(m)$. We then obtain $\frac{\mathrm{d}}{\mathrm{d}t}U_1\big|_{t=0} = (I - UU^\top)ZVS^{-1} + U\mathrm{skew}(U^\top ZV)S^{-1}$, $\frac{\mathrm{d}}{\mathrm{d}t}\hat{S}_1\Big|_{t=0} = \mathrm{sym}(U^\top ZV)$, $\frac{\mathrm{d}}{\mathrm{d}t}\tilde{S}_0\big|_{t=0} = \mathrm{sym}(U^\top ZV) - U^\top ZV = -\mathrm{skew}(U^\top ZV)$, $\frac{\mathrm{d}}{\mathrm{d}t}V_1\big|_{t=0} = (I - VV^\top)(V\mathrm{skew}(U^\top ZV) + Z^\top U)S^{-T} + V\mathrm{skew}(V^\top(V\mathrm{skew}(U^\top ZV) + Z^\top U))S^{-T} = (I - VV^\top)Z^\top US^{-T}$, and $\frac{\mathrm{d}}{\mathrm{d}t}S_1^\top\big|_{t=0} = \mathrm{sym}(V^\top(V\mathrm{skew}(U^\top ZV) + Z^\top U)) = \mathrm{sym}(V^\top SU)$. This yields $\frac{\mathrm{d}}{\mathrm{d}t}U_1S_1V_1^\top\big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}U_1\big|_{t=0}SV^\top + U\,\frac{\mathrm{d}}{\mathrm{d}t}S_1\big|_{t=0}V^\top + US\,\frac{\mathrm{d}}{\mathrm{d}t}V_1^\top\big|_{t=0} = \mathcal{P}_X Z$.

We have thus shown the following.

Proposition 3.1 (Lie–Trotter extended retraction)**.** *The R mapping defined by* (24) *is a well-defined extended retraction on $\mathcal{M}_r$.*

Note that, since U_1 appears instead of U in (22) and (23), the component of Z in the normal space $T_X^\perp\mathcal{M}_r$ has an impact on the outcome (24); that is, $R(X,Z) \neq R(X,\mathcal{P}_X Z)$ in general.

We now proceed to show that the Lie–Trotter extended retraction is a second-order retraction. The proof will make use of the following remarkable property of the Lie–Trotter extended retraction, which can be deduced directly from [LO13, Theorem 4.1].

Proposition 3.2 (exactness property)**.** *Let R be the Lie–Trotter extended retraction defined in* (24)*. For all $X, Y \in \mathcal{M}_r$, it holds that $R(X, Y - X) = Y$.*

Note that the expression $R(X, Y - X)$ only makes sense for extended retractions, as $Y - X$ does in general not belong to $\mathrm{T}_X\mathcal{M}_r$.

Theorem 3.3 (second-order property)**.** *Let R be the Lie–Trotter extended retraction defined in* (24)*. For all $X \in \mathrm{T}_X\mathcal{M}_r$ and all $\dot{X} \in \mathrm{T}_X\mathcal{M}_r$, it holds that $R(X, t\dot{X}) = R_{ortho}(X, t\dot{X}) + O(t^3)$. Since R_{ortho} is a second-order retraction, it follows that R is a second-order retraction.*

Proof. We freely drop the foot X in the notation. Observe that $R_{\mathtt{ortho}}(t\dot{X}) = X + t\dot{X} + O_{\mathrm{N}}(t^2)$ where $O_{\mathrm{N}}(t^2) \in T_X^\perp\mathcal{M}_r$ with $\|O_{\mathrm{N}}(t^2)\| \leq ct^2$ for all t sufficiently small. This yields

$$\begin{aligned}
R(t\dot{X}) &= R(R_{\mathtt{ortho}}(t\dot{X}) - X + O_{\mathrm{N}}(t^2)) \\
&= R(R_{\mathtt{ortho}}(t\dot{X}) - X) + \mathrm{D}R(R_{\mathtt{ortho}}(t\dot{X}) - X)[O_{\mathrm{N}}(t^2)] + O(t^4) \\
&= R(R_{\mathtt{ortho}}(t\dot{X}) - X) + (\mathrm{D}R(0) + O(t))[O_{\mathrm{N}}(t^2)] + O(t^4) \\
&= R(R_{\mathtt{ortho}}(t\dot{X}) - X) + O(t^3)
\end{aligned}$$

since $\mathrm{D}R(0)$ is the projection onto the tangent space,

$$= R_{\mathtt{ortho}}(t\dot{X}) + O(t^3)$$

by the exactness property. ∎

3.7.2 Flop count

Assuming that Z is a full $m \times n$ matrix and X is available in the factorized form (5), the dominant cost is $2mr^2 + 2mnr[US + ZV] + 4mr^2[U_1, \hat{S}_1] + 2mr^2[U_1^\top ZV] + r^2[\hat{S}_1 - U_1^\top ZV] + 2nr^2 + 2mnr[V\tilde{S}_0^\top + Z^\top U_1] + 4nr^2[V_1, S_1^\top]$, hence $4mnr + 8mr^2 + 6nr^2$.

Note however that this extended retraction does not play in the same league as the other retractions mentioned above, as its input is a full $m \times n$ matrix Z instead of an $\dot{X}$ represented as in (12). Recall from Section 2.2 that computing $\dot{X}$ as $\mathcal{P}_X Z$ already requires $4mnr + 4mr^2 + 2nr^2$ flops.

The dominant additional flop count to compute $R(X, tZ)$ for a new value of t is $4mr^2[U_1, \hat{S}_1] + 2mr^2[tU_1^\top ZV] + r^2[\hat{S}_1 - tU_1^\top ZV] + 2mnr[V\tilde{S}_0^\top + tZ^\top U_1] + 4nr^2[V_1, S_1^\top]$. In comparison, for each retraction $\tilde{R}$ defined above, computing $\tilde{R}(X, t\mathcal{P}_X Z)$ for a new t has a cost of $O((m+n)r^2)$ only, since $\mathcal{P}_X Z$ can be precomputed. The Lie–Trotter extended retraction is thus not competitive in the "new t" scenario.

3.8 Lie–Trotter retraction

The (KSL) Lie–Trotter retraction R is simply defined as the Lie–Trotter extended retraction (24) where Z is restricted to belong to $\mathrm{T}_X\mathcal{M}_r$. The sole purpose of this section is to present a computationally efficient way of computing $R(X, \dot{X})$ when $\dot{X}$ is available in the (U_p, S, V_p) form (12).

1. Get U_1 and $\hat{S}_1$ by an orthonormalization $U_1\hat{S}_1 = U(S + \dot{S}) + U_p$ with U_1 orthonormal.
2. Get $\tilde{S}_0$ by $\tilde{S}_0 = \hat{S}_1 - (U_1^\top U_p + (U_1^\top U)\dot{S})$.
3. Get V_1 and S_1 by an orthonormalization $V_1 S_1^\top = V\tilde{S}_0^\top + Z^\top U_1$, with Z as in (12).

Finally, set
$$R(X, \dot{X}) = U_1 S_1 V_1^\top.$$

This is a second-order retraction in view of the analysis in Section 3.7.

3.8.1 Matlab implementation details

This retraction is labeled `KSL`. We use the polar decomposition for the orthonormalizations. From the first step, we have $U^\top U_1 = (S + \dot{S})\hat{S}_1^{-1}$. We compute $V\tilde{S}_0^\top + Z^\top U_1$ as $V(\tilde{S}_0^\top + U_p^\top U_1 + \dot{S}^\top(U^\top U_1)) + V_p(U^\top U_1)$.

3.8.2 Flop count

The dominant flop count is $2mr^2[U(S + \dot{S}) + U_p] + 4mr^2[U_1, \hat{S}_1] + O(r^3)[U^\top U_1] + 2mr^2[U_1^\top U_p] + 4nr^2[V\tilde{S}_0^\top + Z^\top U_1] + 4nr^2[V_1, S_1^\top]$, that is, $8(m+n)r^2$.

The dominant flop count to compute $R(X, t\dot{X})$ for a new t can be reduced to $4mr^2 + 6nr^2$.

3.9 Modified Lie–Trotter retraction

Observe that the above procedure adds $\dot{S}$ twice and subtracts it once. Instead, one could modify the procedure as follows to add it only once:

1. Get U_1 and $\hat{S}_1$ by a decomposition $U_1\hat{S}_1 = US + U_p$ with U_1 orthonormal.

2. Get $\tilde{S}_0$ by $\tilde{S}_0 = \hat{S}_1 + U_1^\top U\dot{S}$.

3. Get V_1 and S_1 by a decomposition $V_1 S_1^\top = V\tilde{S}_0^\top + V_p U^\top U_1$.

Finally, set

$$R(X, \dot{X}) = U_1 S_1 V_1^\top.$$

This can be shown to be a retraction using techniques similar to those employed in Section 3.7. However, numerical experiments indicate that this is not a second-order retraction; see Section 4.2.

3.9.1 Matlab implementation details

This retraction is labeled `KSL+`. In view of the first step, the product $U^\top U_1$ appearing in steps 2 and 3 can be computed as $S\hat{S}_1^{-1}$.

3.9.2 Flop count

The dominant flop count is $2mr^2[US + U_p] + 4mr^2[U_1, \hat{S}_1] + 4nr^2[V\tilde{S}_0^\top + V_p U^\top U_1] + 4nr^2[V_1, S_1^\top]$, thus $6mr^2 + 8nr^2$. When $m < n$, the "LSK" way (computing V_1 first) is preferable in terms of flops; this amounts to the (different) retraction that maps $(X, \dot{X})$ to $R(X^\top, \dot{X}^\top)^\top$, with dominant flop count of $8mr^2 + 6nr^2$. We use the LSK way in our experiments when $m < n$.

The dominant flop count to compute $R(X, t\dot{X})$ for a new t can be reduced to $2mr^2 + 6nr^2$ if adequate matrix products are precomputed.

3.10 Exponential retraction

The exponential retraction is defined by $R(X, \dot{X}) = \gamma(1)$, where γ is the geodesic on $\mathcal{M}_r$ (viewed as a Riemannian submanifold of $\mathbb{R}^{m\times n}$) with initial conditions $\gamma(0) = X$ and $\gamma'(0) = \dot{X}$. The exponential is arguably the "theoretically ideal" retraction, but it was realized early on that trying to move along geodesics is usually computationally expensive [Lue72]. Formalizing the idea of resorting instead to first-order approximations of geodesics was a motivation behind the concept of retraction [Shu86, ADM+02]. And indeed, for our manifold $\mathcal{M}_r$, we are not aware of a closed-form expression for the geodesics. (Note however that there is a Riemannian metric on $\mathcal{M}_r$, different from the embedded metric, for which geodesics admit a closed-form expression; see [AAM13, §6.11].)

Nevertheless, it is worthwhile investigating how the retractions proposed above compare with the exponential. To this end, we have implemented the following basic numerical scheme for solving the geodesic equation $\mathcal{P}_{X(t)}X''(t) = 0$, where $\mathcal{P}$ is the tangent projection (9): $X(t+\delta) = R_{\texttt{ortho}}(X(t), \delta\dot{X}(t))$, $\dot{X}(t+\delta) = \mathcal{P}_{X(t+\delta)}\dot{X}(t)$. As usually in numerical integration schemes, the choice of δ is guided by the conflicting goals of achieving low truncation errors, rounding errors,

Table 1: Pairwise distances with $m = 1.0e+03, n = 1.0e+03, r = 1.0e+01, t = 1.0e-04$.

	proj	ortho	StRSt_pj	RRR	ez-2nd	Shalit	KSL	KSL+	geod
proj	0.0e+00	1.3e-05	1.8e-04	1.9e-05	1.3e-05	1.3e-05	9.2e-06	1.6e-05	8.8e-06
ortho	1.3e-05	0.0e+00	1.8e-04	2.1e-05	3.4e-08	1.6e-08	9.5e-06	1.9e-05	4.4e-06
StRSt_pj	1.8e-04	1.8e-04	0.0e+00	1.7e-04	1.8e-04	1.8e-04	1.8e-04	1.7e-04	1.8e-04
RRR	1.9e-05	2.1e-05	1.7e-04	0.0e+00	2.1e-05	2.1e-05	2.0e-05	1.1e-05	2.0e-05
ez-2nd	1.3e-05	3.4e-08	1.8e-04	2.1e-05	0.0e+00	2.3e-08	9.5e-06	1.9e-05	4.4e-06
Shalit	1.3e-05	1.6e-08	1.8e-04	2.1e-05	2.3e-08	0.0e+00	9.5e-06	1.9e-05	4.4e-06
KSL	9.2e-06	9.5e-06	1.8e-04	2.0e-05	9.5e-06	9.5e-06	0.0e+00	1.8e-05	7.0e-06
KSL+	1.6e-05	1.9e-05	1.7e-04	1.1e-05	1.9e-05	1.9e-05	1.8e-05	0.0e+00	1.7e-05
geod	8.8e-06	4.4e-06	1.8e-04	2.0e-05	4.4e-06	4.4e-06	7.0e-06	1.7e-05	0.0e+00

and computation times. In our numerical experiments, we found that $\delta = 10^{-3}$ was an acceptable compromise, and we did not attempt to choose δ adaptively.

This retraction is labeled `geod`.

4 Experiments

We now compare numerically the various retractions described above. The Matlab code that generated the tables is available from `http://sites.uclouvain.be/absil/2013.04`.

4.1 Pairwise distances

In a first set of experiments reported in Table 1, we compute the pairwise distances $\|R_i(X, t\dot{X}) - R_j(X, t\dot{X})\|_F$, where R_i stands for the ith retraction in our list. Matrix X is represented as in (5), where U and V are generated by orthonormalizing matrices drawn from the standard normal distribution and S is drawn from the standard normal distribution. The tangent vector $\dot{X} = \mathcal{P}_X Z$ is generated by drawing an $m \times n$ matrix Z from the standard normal distribution. For this small value of t, one observes that the second-order retractions R (i.e., all the retractions but `StRSt_pj`, `RRR`, and `KSL+`) achieve the smaller distance between $R(X, t\dot{X})$ and $R_{\texttt{geod}}(X, t\dot{X})$.

Table 2 shows the results obtained for the same experiment but with S now chosen to have a large condition number. Specifically, we choose S with singular values equal to 1 ($r-1$ times) and 10^{-6} (one time). The various retractions are seen to behave very differently in this ill-conditioned setting. In particular, a large discrepancy is observed between `RRR`, `ez-2nd`, `Shalit`, and the other retractions. These retractions are readily seen to be unbounded: bounded inputs do not yield bounded outputs. The `orth` retraction is also unbounded, but the unboundedness becomes apparent for inputs such that $S + \dot{S}$ is ill-conditioned. All the other retractions considered above are bounded, namely `proj`, `StRSt_pj`, `KSL`, `KSL+`, and `geod`.

Several other experiments on pairwise distances could be conducted to get a more detailed understanding of the differences between the various retractions. In particular, since the retractions do not have the same domain of definition, pushing them to the limit of their domain of definition can reveal marked differences.

Table 2: Pairwise distances with $m = \texttt{1.0e+03}, n = \texttt{1.0e+03}, r = \texttt{1.0e+01}, t = \texttt{1.0e-04}$.

	proj	ortho	StRSt_pj	RRR	ez-2nd	Shalit	KSL	KSL+	geod
proj	0.0e+00	9.4e-02	3.2e-03	1.0e+03	1.0e+03	2.5e+04	2.2e-03	3.8e-03	1.9e-03
ortho	9.4e-02	0.0e+00	9.6e-02	1.0e+03	1.0e+03	2.5e+04	9.6e-02	9.6e-02	9.5e-02
StRSt_pj	3.2e-03	9.6e-02	0.0e+00	1.0e+03	1.0e+03	2.5e+04	3.3e-03	3.2e-03	4.0e-03
RRR	1.0e+03	1.0e+03	1.0e+03	0.0e+00	2.0e+03	2.4e+04	1.0e+03	1.0e+03	1.0e+03
ez-2nd	1.0e+03	1.0e+03	1.0e+03	2.0e+03	0.0e+00	2.6e+04	1.0e+03	1.0e+03	1.0e+03
Shalit	2.5e+04	2.5e+04	2.5e+04	2.4e+04	2.6e+04	0.0e+00	2.5e+04	2.5e+04	2.5e+04
KSL	2.2e-03	9.6e-02	3.3e-03	1.0e+03	1.0e+03	2.5e+04	0.0e+00	4.5e-03	3.7e-03
KSL+	3.8e-03	9.6e-02	3.2e-03	1.0e+03	1.0e+03	2.5e+04	4.5e-03	0.0e+00	3.9e-03
geod	1.9e-03	9.5e-02	4.0e-03	1.0e+03	1.0e+03	2.5e+04	3.7e-03	3.9e-03	0.0e+00

Table 3: Norm of the tangent projection of the second-order finite difference.

t	proj	ortho	StRSt_pj	RRR	ez-2nd	Shalit	KSL	KSL+	geod
1.0e-03	1.2e+01	2.5e-09	1.8e+03	5.2e+02	2.1e-09	1.6e-03	1.9e+00	1.8e+02	2.1e+00
1.0e-04	1.2e-01	2.6e-07	1.8e+03	5.2e+02	2.1e-07	1.6e-05	1.9e-02	1.8e+02	2.2e-02

4.2 Second-order property

As we have seen, retractions `proj`, `ortho`, `ez-2nd`, `Shalit`, `KSL`, and `geod` are second-order retractions. In Table 3, we report an experiment that corroborates this finding and indicates that the other retractions considered above are not second-order retractions. The table provides the values of the Frobenius norm of the tangent projection of the second-order finite difference, i.e., $\delta_i(t) := \|\mathcal{P}_X(R_i(X, t\dot{X}) - 2X + R_i(X, -t\dot{X}))\|_F / t^2$ where R_i stands for the ith retraction in the list. If $\delta_i(t)$ behaves like $O(t)$, it indicates that R_i is a second-order retraction. If it behaves like $O(1)$, it indicates that R_i is not a second-order retraction. It appears from Table 3 that `StRSt_pj`, `RRR`, and `KSL+` are not second-order retractions, while all the other retractions are second-order retractions.

Note that $\delta(t)$ is in fact identically zero in exact arithmetic for `ortho` and `ez-2nd`. This explains why the $O(t)$ behavior is not visible in Table 3 for these two retractions. The property that $\delta(t) \equiv 0$ is obvious for `ortho`: in view of (14), we have that $\mathcal{P}_X(R_{\texttt{ortho}}(X, \dot{X}) - X) = \dot{X}$, hence $\mathcal{P}_X(R_{\texttt{ortho}}(X, t\dot{X}) - 2X + R_{\texttt{ortho}}(X, -t\dot{X})) = \mathcal{P}_X(R_{\texttt{ortho}}(X, t\dot{X}) - X) + \mathcal{P}_X(R_{\texttt{ortho}}(X, -t\dot{X}) - X) = t\dot{X} - t\dot{X} = 0$. For `ez-2nd`, one can show that $\mathcal{P}_X(R_{\texttt{ez-2nd}}(X, t\dot{X}) - X)$ is an odd function of t, and the property follows.

4.3 Symmetry

Most of the retractions presented above are readily seen to preserve symmetry, i.e., $R(X, \dot{X})$ is a symmetric matrix if X and $\dot{X}$ are symmetric matrices. The exceptions are `ez-2nd`, `KSL`, and `KSL+`. The experiments reported in Table 4 confirm that these three retractions do not preserve symmetry. In these experiments, X and $\dot{X}$ are chosen symmetric and the Frobenius norm of $R(X, \dot{X}) - R(X, \dot{X})^\top$ is computed.

Table 4: Symmetry test.

proj	ortho	StRSt_pj	RRR	ez-2nd	Shalit	KSL	KSL+	geod
6.9e-14	1.0e-13	4.9e-14	8.3e-14	2.1e+00	9.9e-14	1.1e+01	1.1e+01	1.3e-12

4.4 Run times

For information, we report wall-clock computation times for X of square shape (Table 5) and for X of horizontal shape (Table 6). The tests were run with MATLAB Version 7.13.0.564 (R2011b) on a PC with two Intel(R) Pentium(R) D CPU 3.00GHz, 2048 KB cache each, running Linux kernel 3.2.0. The timings observed are compatible with the flop counts mentioned for the various retractions, and `geod`, as expected, is much slower.

Table 5: Timing experiments (in seconds) with $m = \texttt{1.0e+04}, n = \texttt{1.0e+04}, r = \texttt{1.0e+01}$.

proj	ortho	StRSt_pj	RRR	ez-2nd	Shalit	KSL	KSL+	geod
2.6e-02	2.3e-02	2.2e-02	2.2e-02	2.1e-02	2.5e-02	2.4e-02	2.3e-02	5.8e+01

Table 6: Timing experiments (in seconds) with $m = \texttt{1.0e+03}, n = \texttt{1.0e+04}, r = \texttt{1.0e+01}$.

proj	ortho	StRSt_pj	RRR	ez-2nd	Shalit	KSL	KSL+	geod
1.5e-02	1.3e-02	1.3e-02	1.3e-02	1.3e-02	1.5e-02	1.5e-02	1.3e-02	3.6e+01

To test how these figures may depend on the platform, we also ran the same experiments with Matlab 8.1.0.604 (R2013a) on a PC with four Intel(R) Xeon(R) CPU X3210 2.13GHz, 4096 KB cache each, running Linux kernel 2.6.18. The outcomes are presented in Tables 7 and (8).

5 Concluding comments

We have presented, analyzed, and tested numerically several retractions on the low-rank manifold $\mathcal{M}_r$ of rank-r $m \times n$ matrices. In the absence of a closed-form expression, the exponential retraction (`geod`) is clearly the least time-efficient one, confirming that much computational effort can be spared by considering other retractions. A rather good coherence has been observed between flop counts and run times, but the differences between the various retractions along these criteria are rather inconsequential, except for `geod`. However, the various retractions differ markedly according to properties such as domain of definition, boundedness, first/second-order property, and symmetry. This work has not allowed a “best” retraction to emerge, but it has provided the developer of low-rank numerical methods with a panoply mechanisms (retractions) to smoothly produce curves of fixed-rank matrices with given initial position and velocity.

Table 7: Timing experiments (in seconds) with $m = \texttt{1.0e+04}, n = \texttt{1.0e+04}, r = \texttt{1.0e+01}$.

proj	ortho	StRSt_pj	RRR	ez-2nd	Shalit	KSL	KSL+	geod
1.3e-02	9.9e-03	1.1e-02	1.0e-02	1.1e-02	1.2e-02	1.2e-02	1.1e-02	2.5e+01

Table 8: Timing experiments (in seconds) with $m = \texttt{1.0e+03}, n = \texttt{1.0e+04}, r = \texttt{1.0e+01}$.

proj	ortho	StRSt_pj	RRR	ez-2nd	Shalit	KSL	KSL+	geod
6.9e-03	5.8e-03	5.9e-03	5.8e-03	5.8e-03	6.8e-03	6.5e-03	6.0e-03	1.5e+01

References

[AAM13] P.-A. Absil, Luca Amodei, and Gilles Meyer. Two Newton methods on the manifold of fixed-rank matrices endowed with riemannian quotient geometries. *Computation. Stat.*, 2013. doi:doi:10.1007/s00180-013-0441-6.

[ABG07] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, 7(3):303–330, July 2007. doi:10.1007/s10208-005-0179-9.

[ADM+02] Roy L. Adler, Jean-Pierre Dedieu, Joseph Y. Margulies, Marco Martens, and Mike Shub. Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA J. Numer. Anal.*, 22(3):359–390, July 2002. doi:10.1093/imanum/22.3.359.

[AM12] P.-A. Absil and Jérôme Malick. Projection-like retractions on matrix manifolds. *SIAM J. Optim.*, 22(1):135–158, 2012. doi:10.1137/100802529.

[AMS08] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. URL: http://press.princeton.edu/titles/8586.html.

[BMAS13] Nicolas Boumal, Bamdev Mishra, P. A. Absil, and Rodolphe Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds, 2013. arXiv:1308.5200.

[Boo03] William M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 2003. Revised Second Edition.

[GV96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[KL07] O. Koch and C. Lubich. Dynamical low-rank approximation. *SIAM Journal on Matrix Analysis and Applications*, 29(2):434–454, 2007. doi:10.1137/050639703.

[KSV13] D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by riemannian optimization. Technical report, 2013.

[Lee03] John M. Lee. *Introduction to smooth manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2003.

[LO13] Ch. Lubich and I.V. Oseledets. A projector-splitting integrator for dynamical low-rank approximation, 2013. `arXiv:1301.1058v2`.

[Lue72] David G. Luenberger. The gradient projection method along geodesics. *Management Sci.*, 18:620–631, 1972.

[MMBS11] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre. Low-rank optimization with trace norm penalty, 2011. Accepted for publication in SIAM Journal on Optimization. `arXiv:1112.2318v1`.

[MMBS13] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre. Fixed-rank matrix factorizations and Riemannian low-rank optimization, 2013. `arXiv:1209.0430v2`.

[MS13] B. Mishra and R. Sepulchre. R3MC: A Riemannian three-factor algorithm for low-rank matrix completion, 2013. `arXiv:1306.2672v1`.

[Ros61] J. B. Rosen. The gradient projection method for nonlinear programming. II. Nonlinear constraints. *J. Soc. Indust. Appl. Math.*, 9:514–532, 1961.

[RW12] Wolfgang Ring and Benedikt Wirth. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM J. Optim.*, 22(2):596–627, 2012. `doi:10.1137/11082885X`.

[Shu86] Michael Shub. Some remarks on dynamical systems and numerical analysis. In L. Lara-Carrero and J. Lewowicz, editors, *Proc. VII ELAM.*, pages 69–92. Equinoccio, U. Simón Bolívar, Caracas, 1986.

[SWC13] Uri Shalit, Daphna Weinshall, and Gal Chechik. Online learning in the embedded manifold of low-rank matrices. *J. of Mach. Learn. Res.*, 13:429–458, 2013.

[Van13] Bart Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM J. Optim.*, 23(2):1214–1236, 2013. `doi:10.1137/110845768`.