

## GRADIENT ALGORITHMS FOR PRINCIPAL COMPONENT ANALYSIS

R. E. MAHONY<sup>1</sup>, U. HELMKE<sup>2</sup> and J. B. MOORE<sup>1</sup>

(Received 28 January 1994; revised 1 July 1994)

### Abstract

The problem of principal component analysis of a symmetric matrix (finding a  $p$ -dimensional eigenspace associated with the largest  $p$  eigenvalues) can be viewed as a smooth optimization problem on a homogeneous space. A solution in terms of the limiting value of a continuous-time dynamical system is presented. A discretization of the dynamical system is proposed that exploits the geometry of the homogeneous space. The relationship between the proposed algorithm and classical methods are investigated.

### 1. Introduction

The problem of principal component analysis of a symmetric matrix  $N = N^T$  is that of finding an eigenspace of specified dimension  $p \geq 1$  which corresponds to the maximal  $p$  eigenvalues of  $N$ . There are a number of classical algorithms available for computing dominant eigenspaces (principal components) of a symmetric matrix. A good reference for standard numerical methods is Golub and Van Loan [10].

There has been considerable interest in the last decade in using dynamical systems to solve linear algebra problems (see the review [5] and the recent monograph [11]). It is desirable to consider the relationship between such methods and classical algebraic methods. For example, Deift *et al.* [6] investigated a matrix differential equation based on the Toda flow, the solution of which (evaluated at integer times) is exactly the sequence of iterates generated by the standard  $QR$  algorithm. In general, dynamical system solutions of linear algebra problems do not interpolate classical methods exactly. Discrete computational methods based on dynamical system solutions to a given problem provide a way of comparing classical algorithms with dynamical system methods. Recent work on developing numerical methods based on dynamical

<sup>1</sup>Dept of Systems Eng, Research School of Phys. Sciences and Eng, ANU, Canberra ACT 0200.

<sup>2</sup>Dept of Mathematics, University of Regensburg, 8400 Regensburg, F.R.G.

© Australian Mathematical Society, 1996, Serial-fee code 0334-2700/96

systems insight is contained Brockett [3] and Moore *et al.* [14].

Concentrating on the problem of principal component analysis, Ammar and Martin [1] have studied the power method (for determining the dominant  $p$ -dimensional eigenspace of a symmetric matrix) as a recursion on the Grassmannian manifold  $G^p(\mathbb{R}^n)$ , the set of all  $p$ -dimensional subspaces of  $\mathbb{R}^n$ . Using local coordinate charts on  $G^p(\mathbb{R}^n)$  Ammar and Martin [1] show that the power method is closely related to the solution of a matrix Riccati differential equation. Unfortunately, the solution to a matrix Riccati equation may diverge to infinity in finite time. Such solutions correspond to solutions that do not remain in the original local coordinate chart. In his review paper, Chu [5] derives the gradient flow of the Rayleigh quotient (for a symmetric matrix) on the sphere. Chu's result is a vector differential equation in  $\mathbb{R}^n$  whose limiting solution is the maximal eigenvector of the matrix considered. It turns out that Chu's result is a simple case of a matrix differential equation proposed by Oja [15, 16] for the analysis of learning performance of single-layer neural networks with  $n$  inputs and  $p$  neurons. The differential equation that Oja considers evolves on  $\mathbb{R}^{n \times p}$  and corresponds to the 'learning' procedure of the neural network. The columns of the limiting solution span the principal component of the covariance matrix  $N = E\{u_k u_k^T\}$  (where  $E\{u_k u_k^T\}$  is the expectation of  $u_k u_k^T$ ) of the vector random process  $u_k \in \mathbb{R}^n$ ,  $k = 1, 2, \dots$ , with which the network was 'trained'. Recent work by Yan *et al.* [18] has provided a rigorous analysis of the learning equation proposed by Oja. Not surprisingly, it is seen that the solution to Oja's learning equations is closely related to the solution of a Riccati differential equation [11, page 27].

In this paper we follow Yan *et al.* [18] and study the properties of Oja's learning equation restricted to the Stiefel manifold (the set of all  $n \times p$  real matrices with orthonormal columns). However, we differ from earlier treatments by considering a homogeneous geometric structure on the Stiefel manifold. Oja's flow is derived as the gradient flow of a generalised Rayleigh quotient and explicit proofs of convergence for the flow are presented which extend the results of Yan *et al.* [18] and Helmke and Moore [11, page 26] so that no genericity assumption is required on the eigenvalues of  $N$ . The homogeneous nature of the Stiefel manifold is exploited to develop an explicit numerical method (a discrete-time system evolving on the Stiefel manifold) for principal component analysis. The method proposed is a gradient ascent algorithm modified to evolve explicitly on  $\text{St}(p, n)$ . A step-size must be selected for each iteration and a suitable selection scheme is proposed. A proof of convergence for the proposed algorithm is given as well as modifications and observations aimed at reducing the computational cost of implementing the algorithm on a digital computer. The discrete method proposed is similar to the classical power method and steepest ascent methods for determining the dominant  $p$ -eigenspace of a matrix  $N$ . Indeed, in the case where  $p = 1$  (for a particular choice of time-step) the discretization is shown to be the power method. When  $p > 1$ , however, there are subtle differences between

the methods.

The paper is organised into five sections including the introduction. Section 2 reviews the derivation of the Oja flow and gives a general proof of convergence. In Section 3 a discrete-time iteration based on the results in Section 2 is proposed, along with a suitable choice of time-step. Section 4 considers two modifications of the scheme to reduce the computational cost of implementing the proposed numerical algorithm. Finally Section 5 considers the relationship of the proposed algorithm to classical methods.

## 2. Continuous-time gradient flow

In this section a dynamical systems solution to the problem of finding the principal component of a matrix is developed, based on computing the gradient flow associated with a generalised Rayleigh quotient function on the Stiefel manifold. The Stiefel manifold is given the structure of a homogeneous space (rather than just an embedded submanifold of  $\mathbb{R}^{n \times p}$ ). This approach differs from the approach previous authors have taken in related work [15, 16, 11, 18], and we take the time to present the geometry before the principal result of this section is presented. The reader is referred to Warner [17] for general technical details on Lie-groups and homogeneous spaces.

Let  $N = N^T$  be a real symmetric  $n \times n$  matrix with eigenvalues  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  and an associated set of orthonormal eigenvectors  $v_1, \dots, v_n$ . A *maximal  $p$ -dimensional eigenspace*, or *maximal  $p$ -eigenspace* of  $N$ , is  $\text{sp}\{v_1, \dots, v_p\}$  the subspace of  $\mathbb{R}^n$  spanned by  $\{v_1, \dots, v_p\}$ . If  $\lambda_p > \lambda_{p+1}$  then the maximal  $p$ -eigenspace of  $N$  is unique. If  $\lambda_p = \lambda_{p+1} = \dots = \lambda_{p+r}$ , for some  $r > 0$ , then any subspace  $\text{sp}\{v_1, \dots, v_{p-1}, w\}$ , where  $w \in \text{sp}\{v_p, v_{p+1}, \dots, v_{p+r}\}$ , is a maximal  $p$ -eigenspace of  $N$ .

For  $p$  an integer with  $1 \leq p \leq n$ , let

$$\text{St}(p, n) = \left\{ X \in \mathbb{R}^{n \times p} \mid X^T X = I_p \right\}, \quad (1)$$

where  $I_p$  is the  $p \times p$  identity matrix, denote the *Stiefel manifold* of real orthogonal  $n \times p$  matrices. For  $X \in \text{St}(p, n)$ , the columns of  $X$  are orthonormal basis vectors for a  $p$ -dimensional subspace of  $\mathbb{R}^n$ . The *Stiefel manifold*  $\text{St}(p, n)$  is a smooth compact  $np - \frac{1}{2}p(p+1)$ -dimensional submanifold of  $\mathbb{R}^{n \times p}$  [11, page 25]. The proof given in Helmke and Moore [11] exploits the fact that  $I_p$  is a regular point of the map  $X \mapsto X^T X$ . One can also think of  $\text{St}(p, n)$  as a homogeneous space and it is this property of  $\text{St}(p, n)$  that is exploited later to develop a numerical method.

Let  $G = O(n) \times O(p)$  be the topological product of the set of  $n \times n$  and  $p \times p$  real orthogonal matrices  $O(n) = \{U \in \mathbb{R}^n \mid U^T U = U U^T = I_n\}$ . Then  $G$  is a compact

Lie-group [11, page 348]. It is easily verified that  $\gamma : G \times \text{St}(p, n) \rightarrow \text{St}(p, n)$  given by

$$\gamma((U, V), X) := UXV^T \tag{2}$$

is a smooth, transitive, group action of  $G$  on  $\text{St}(p, n)$ . Since  $G$  is compact it follows that  $\text{St}(p, n)$  is a compact embedded submanifold of  $\mathbb{R}^{n \times p}$  [11, page 352]. The tangent space  $T_X \text{St}(p, n)$  of  $\text{St}(p, n)$  at a point  $X \in \text{St}(p, n)$  is given by the image of the linearization of  $\gamma_X : G \rightarrow \text{St}(p, n)$ ,

$$\gamma_X(U) := \gamma(U, X),$$

at the identity element of  $G$  [9, page 75]. Recall that the tangent space of  $O(n)$  at the identity is [11, page 349]

$$T_n O(n) = Sk(n) = \left\{ \Omega \in \mathbb{R}^{n \times n} \mid \Omega^T = -\Omega \right\},$$

and consequently that the tangent space at the identity of  $G$  is  $T_{(I_n, I_p)}G = Sk(n) \times Sk(p)$ . It follows that

$$T_X \text{St}(p, n) = \{ \Omega X - X \Pi \mid \Omega \in Sk(n), \Pi \in Sk(p) \}. \tag{3}$$

The natural Riemannian metric to use with the homogeneous structure of  $\text{St}(p, n)$  is the normal metric derived from a right invariant metric on the Lie-group  $G$  [13, page 127]. To construct this metric consider the Euclidean inner product on  $\mathbb{R}^{n \times n} \times \mathbb{R}^{p \times p}$  as

$$\langle (\Omega_1, \Pi_1), (\Omega_2, \Pi_2) \rangle = \text{tr}(\Omega_1^T \Omega_2) + \text{tr}(\Pi_1^T \Pi_2).$$

This induces a nondegenerate inner product on  $T_{(I_n, I_p)}G$ . Given  $X \in \text{St}(p, n)$ , then the linearization  $T_{(I_n, I_p)}\gamma_X$  of  $\gamma_X$  can be used to decompose the identity tangent space into

$$T_{(I_n, I_p)}G = \ker T_{(I_n, I_p)}\gamma_X \times \text{dom } T_{(I_n, I_p)}\gamma_X,$$

where  $\ker T_{(I_n, I_p)}\gamma_X$  is the kernel of  $T_{(I_n, I_p)}\gamma_X$  and

$$\text{dom } T_{(I_n, I_p)}\gamma_X = \{ (\Omega_1, \Pi_1) \in T_{(I_n, I_p)}G \mid \langle (\Omega_1, \Pi_1), (\Omega, \Pi) \rangle = 0, (\Omega, \Pi) \in \ker T_{(I_n, I_p)}\gamma_X \}$$

is the domain of  $T_{(I_n, I_p)}\gamma_X$  (the subspace orthogonal to  $\ker T_{(I_n, I_p)}\gamma_X$  using the Euclidean inner product provided on  $T_{(I_n, I_p)}G$ ). By construction,  $T_{(I_n, I_p)}\gamma_X$  restricts to a vector space isomorphism  $T_{(I_n, I_p)}^\perp \gamma_X$ ,

$$\begin{aligned} T_{(I_n, I_p)}^\perp \gamma_X &: \text{dom } T_{(I_n, I_p)}\gamma_X \rightarrow T_X \text{St}(p, n), \\ T_{(I_n, I_p)}^\perp \gamma_X(X) &:= T_{(I_n, I_p)}\gamma_X(X). \end{aligned}$$

The normal Riemannian metric (see Helmke and Moore [11, page 52]) on  $\text{St}(p, n)$  is the nondegenerate bilinear map on each tangent space  $T_X \text{St}(p, n)$

$$\langle (\Omega_1 X - X \Pi_1, \Omega_2 X - X \Pi_2) \rangle_X = \text{tr}((\Omega_1^\perp)^\top \Omega_2^\perp) + \text{tr}((\Pi_1^\perp)^\top \Pi_2^\perp), \tag{4}$$

where  $\Omega_i X - X \Pi_i \in T_X \text{St}(p, n)$  for  $i = 1, 2$  and

$$(\Omega_i, \Pi_i) = ((\Omega_i)_{\perp}, (\Pi_i)_{\perp}) \oplus (\Omega_i^{\perp}, \Pi_i^{\perp})$$

is the decomposition of  $(\Omega_i, \Pi_i)$  into components in  $\ker T_{(I_n, I_p)} \gamma_X$  and  $\text{dom } T_{(I_n, I_p)} \gamma_X$  respectively. It is easily verified that  $\langle \langle \cdot, \cdot \rangle \rangle_X$  varies smoothly with  $X$  and defines a Riemannian metric.

The manifold  $\text{St}(p, n)$  provides a smooth constraint set on which the problem of principal component analysis may be considered. In the case  $p = 1$  (where only a single maximal eigenvector is desired) one may consider optimising the cost index  $r_N : \mathbb{R}^n - \{0\} \rightarrow \mathbb{R}$ ,

$$r_N(x) = \frac{x^\top N x}{x^\top x},$$

known as the classical Rayleigh quotient. Of course when  $x \in \text{St}(1, n)$  then  $r_N(x) = x^\top N x$ . The generalised Rayleigh quotient on  $\text{St}(p, n)$  is defined as

$$R_N : \text{St}(p, n) \rightarrow \mathbb{R}, \quad R_N(X) = \text{tr}(X^\top N X). \tag{5}$$

To confirm that optimizing  $R_N$  on  $\text{St}(p, n)$  provides a solution to the problem of principal component analysis, recall the Ky-Fan minimax principle [12, page 191], which states

$$\begin{aligned} \max_{X \in \text{St}(p, n)} R_N(X) &= \lambda_1 + \dots + \lambda_p, \\ \min_{X \in \text{St}(p, n)} R_N(X) &= \lambda_{n+1-p} + \dots + \lambda_n. \end{aligned}$$

Moreover, if  $X \in \text{St}(p, n)$ , such that  $R_N(X) = \sum_{j=1}^p \lambda_j$ , then the columns of  $X$  will generate a basis for a maximal  $p$ -dimensional eigenspace of  $N$ .

We proceed by computing the gradient associated with  $R_N$  on  $\text{St}(p, n)$  and showing that the limiting solution of the continuous-time gradient ascent differential equation converges to a maximum of  $R_N$ .

**THEOREM 2.1.** *Let  $N = N^\top$  be a real symmetric  $n \times n$  matrix and  $p$  an integer with  $1 \leq p \leq n$ . Denote the eigenvalues of  $N$  by  $\lambda_1 \geq \dots \geq \lambda_q$  with algebraic multiplicities  $n_1, \dots, n_q$  such that  $\sum_{i=1}^q n_i = n$ . For  $X \in \text{St}(p, n)$ , define the generalised Rayleigh quotient  $R_N : \text{St}(p, n) \rightarrow \mathbb{R}$ ,  $R_N(X) = \text{tr}(X^\top N X)$ . Then we have the following.*

- (i) *The gradient of  $R_N(X)$ , on the Stiefel manifold  $\text{St}(p, n)$ , with respect to the normal Riemannian metric (4), is*

$$\text{grad } R_N(X) = (I_n - XX^T)NX = [XX^T, N]X \tag{6}$$

(where the Lie bracket of two matrices  $A, B \in \mathbb{R}^{n \times n}$  is  $[A, B] := AB - BA$ ).

- (ii) *The critical points of  $R_N(X)$  on  $\text{St}(p, n)$  are characterised by*

$$[XX^T, N] = 0$$

*and correspond to points  $X \in \text{St}(p, n)$ , such that the columns of  $X$  span a  $p$ -dimensional eigenspace of  $N$ .*

- (iii) *For all initial conditions  $X_0 \in \text{St}(p, n)$ , the solution  $X(t) \in \text{St}(p, n)$  of*

$$\begin{aligned} \frac{d}{dt}X &= \text{grad } R_N(X) \\ &= (I_n - XX^T)NX, \quad X(0) = X_0 \end{aligned} \tag{7}$$

*exists for all  $t \in \mathbb{R}$  and converges to some matrix  $X_\infty \in \text{St}(p, n)$  as  $t \rightarrow \infty$ . For almost all initial conditions the solution  $X(t)$  of (7) converges exponentially fast to a matrix whose columns form a basis for the maximal  $p$ -eigenspace of  $N$ .*

- (iv) *When  $p = 1$  the exact solution to (7) is given by*

$$x(t) = e^{tN}x_0 / \|e^{tN}x_0\|, \tag{8}$$

*where  $x_0 \in S^{n-1} = \text{St}(1, n)$ .*

**PROOF.** The gradient of  $R_N$  is computed using the identities [11, page 356]

- i)  $DR_N|_X(\xi) = \langle \langle \text{grad } R_N(X), \xi \rangle \rangle_X, \quad \xi \in T_X \text{St}(p, n)$
- ii)  $\text{grad } R_N(X) \in T_X \text{St}(p, n)$ ,

where  $DR_N|_X(\xi)$  is the Fréchet derivative of  $R_N(X)$  in direction  $\xi \in T_X \text{St}(p, n)$  evaluated at the point  $X \in \text{St}(p, n)$ . Computing the Fréchet derivative of  $R_N$  in direction  $\Omega X - X\Pi \in T_X \text{St}(p, n)$  gives

$$\begin{aligned} DR_N|_X(\Omega X - X\Pi) &= 2 \text{tr}(X^T N(\Omega X - X\Pi)) \\ &= 2 \text{tr}(XX^T N\Omega) - 2 \text{tr}(X^T N X\Pi). \end{aligned}$$

Observe that  $\text{tr}(X^T N X \Pi) = 0$  since  $X^T N X$  is symmetric and  $\Pi$  is skew symmetric. Similarly only the skew symmetric part of  $X X^T N$  contributes to  $\text{tr}(X X^T N \Omega)$ . Thus,

$$\begin{aligned} DR_N|_X(\Omega X - X \Pi) &= \text{tr}([X X^T, N] \Omega) \\ &= \langle\langle [N, X X^T] X, \Omega X - X \Pi \rangle\rangle_X, \end{aligned}$$

using the Riemannian metric (4). The second line follows since any component of  $[N, X X^T]$  that lies in  $\ker T_{(t,p)} \gamma_X$  does not contribute to the value of  $\text{tr}([X X^T, N] \Omega)$  and of course  $[N, X X^T] \in Sk(n)$  which ensures  $[N, X X^T] X \in T_X \text{St}(p, n)$ . This proves part (i).

At critical points of  $R_N$  the gradient  $\text{grad } R_N(X)$  is zero,  $[N, X X^T] X = 0$ . It follows that  $NX = X(X^T N X)$  and thus the columns of  $X \in \text{St}(p, n)$  span a  $p$ -eigenspace of  $N$ . Moreover  $[N, X X^T] = X(X^T N X)X^T - X(X^T N X)X^T = 0$ , which proves part (ii).

Infinite-time existence of solutions to (7) follows from the compact nature of  $\text{St}(p, n)$ . By applying La'Salle's invariance principle, it is easily verified that  $X(t)$  converges to a level set of  $R_N$  for which  $\text{grad } R_N(X) = 0$ . These sets are termed *critical sets*. To show convergence to a single point (rather than just to a critical set) requires a little extra effort.

An important property of the critical sets of  $R_N$  is that they are a disjoint union of smooth closed manifolds. An explicit proof of this result is contained in [13]; it also follows from the more general development given in [7]. Given  $X \in \text{St}(p, n)$  a critical point of  $R_N$  let  $\mathcal{L}_X \subseteq \text{St}(p, n)$  denote the critical set containing  $X$ . Since  $\mathcal{L}_X$  is a submanifold of  $\text{St}(p, n)$  it has a well defined tangent space,

$$T_X \mathcal{L}_X = \{ \Omega X - X \Pi \mid \Omega \in Sk(n), \Pi \in Sk(p), \text{ and } [N, [\Omega, X X^T]] = 0 \}, \tag{9}$$

at the point  $X \in \text{St}(p, n)$ . To see this, observe that any curve  $Y(t)$ ,  $Y(0) = X$ , lying in  $\mathcal{L}_X$  will satisfy  $[N, Y(t)Y(t)^T]Y(t) = 0$ . Similarly it is easily verified that any curve (passing through  $Y(0) = X$ ) satisfying this equality must lie in  $\mathcal{L}_X$ . Setting  $\dot{Y}(0) = \Omega X - X \Pi \in T_X \text{St}(p, n)$  and then evaluating

$$\frac{d}{dt} [N, Y(t)Y(t)^T]Y(t) = 0$$

gives the above algebraic characterisation for  $T_X \mathcal{L}_X \subset T_X \text{St}(p, n)$ .

Now at a critical point  $X \in R_N$ , the Hessian  $\mathcal{H} R_N$  is a well-defined bilinear map from  $T_X \text{St}(p, n)$  to the reals [11, page 344]. Let  $(\Omega_1 X - X \Pi_1) \in T_X \text{St}(p, n)$  and  $(\Omega_2 X - X \Pi_2) \in T_X \text{St}(p, n)$  be arbitrary; then

$$\begin{aligned} \mathcal{H} R_N(\Omega_1 X - X \Pi_1, \Omega_2 X - X \Pi_2) &= D_{\Omega_1 X - X \Pi_1} (D_{\Omega_2 X - X \Pi_2} R_N(X)) \\ &= D_{\Omega_1 X - X \Pi_1} \text{tr}([X X^T, N] \Omega_2) \\ &= \text{tr}([\Omega_1, X X^T], N] \Omega_2). \end{aligned}$$

Observe that  $[\Omega_1, XX^T]$  is skew symmetric since  $XX^T$  is symmetric and  $\Omega_1$  is skew symmetric. Similarly,  $[[\Omega_1, XX^T], N]$  is skew symmetric. Since  $\Omega_1$  and  $\Omega_2$  are arbitrary then  $\mathcal{H}R_N$  is degenerate in exactly those tangent directions  $(\Omega X - X\Pi) \in T_X \text{St}(p, n)$  for which  $[[\Omega, XX^T], N] = 0$ . But this corresponds exactly to (9) and one concludes that the Hessian  $\mathcal{H}R_N$  degenerates only on the tangent space of the critical set  $\mathcal{L}_X$ . The above argument shows that  $R_N$  is a *Morse-Bott* function on  $\text{St}(p, n)$  [11, page 361]. Applying Proposition C.12.3 from Appendix C [11] completes the proof of part (iii).

Part (iv) of the theorem is verified by explicitly evaluating the derivative of (8).

REMARK 2.2. In the case  $1 < p \leq n$  no exact solution to (7) is known; however, for  $X(t)$  a solution to (7) the solution for  $H(t) = X(t)X(t)^T$  is known, since

$$\begin{aligned} \dot{H}(t) &= \dot{X}X^T + X\dot{X}^T \\ &= NX X^T + XX^T N - 2XX^T N X X^T \\ &= NH(t) + H(t)N - 2H(t)NH(t), \end{aligned} \tag{10}$$

$H(0) = X_0 X_0^T$  and this equation is a Riccati differential equation [18].

### 3. A gradient ascent algorithm

In this section a numerical algorithm for solving (7) is proposed. The algorithm is based on a gradient ascent algorithm modified to ensure that each iteration lies in  $\text{St}(p, n)$ .

Let  $X_0 \in \text{St}(p, n)$  and consider the recursive algorithm generated by

$$X_{k+1} = e^{-\alpha_k [X_k X_k^T, N]} X_k, \tag{11}$$

for a sequence of positive real numbers  $\alpha_k$ , termed *time-steps*. The algorithm generated by (11) is referred to as the *Rayleigh gradient algorithm*. The Lie-bracket  $[X_k X_k^T, N]$  is skew symmetric and consequently  $e^{-\alpha_k [X_k X_k^T, N]}$  is orthogonal and  $X_{k+1} \in \text{St}(p, n)$ . Observe also that

$$\left. \frac{d}{d\tau} e^{-\tau [X_k X_k^T, N]} X_k \right|_{\tau=0} = (I_n - X_k X_k^T) N X_k = \text{grad } R_N(X_k),$$

the gradient of  $R_N$  at  $X_k$ . Thus,  $e^{-\tau [X_k X_k^T, N]} X_k$  represents a curve in  $\text{St}(p, n)$ , passing through  $X_k$  at time  $\tau = 0$ , and with first derivative equal to  $\text{grad } R_N(X)$ . The linearization of  $X_{k+1}(\tau) = e^{-\tau [X_k X_k^T, N]} X_k$  around  $\tau = 0$  is

$$X_{k+1}(\tau) = X_k + \tau \text{grad } R_N(X_k) + (\text{higher order terms}).$$

The higher order terms modify the basic gradient ascent algorithm on  $\mathbb{R}^{n \times p}$  to ensure that the interpolation occurs along curves in  $\text{St}(p, n)$ . For suitably small time-steps  $\alpha_k$ , it is clear that (11) will closely approximate the gradient ascent algorithm on  $\mathbb{R}^{n \times p}$ .

To implement the Rayleigh gradient algorithm it is necessary to choose a time-step  $\alpha_k$ , for each step of the recursion. A convenient criteria for determining suitable time-steps is to maximise the change in potential

$$\Delta R_N(X_k, \alpha_k) = R_N(X_{k+1}) - R_N(X_k). \tag{12}$$

It is possible to use line-search techniques to determine the optimal time-step for each iteration of the algorithm. Completing a line search at each step of the iteration, however, is computationally expensive and often results in worse stability properties for the overall algorithm. Instead, a simple deterministic formulae for the time-step based on maximising a lower bound  $\Delta R'_N(X_k, \tau)$  for (12) is provided.

LEMMA 3.1. *For any  $X_k \in \text{St}(p, n)$  such that  $\text{grad } R_N(X_k) \neq 0$ , the recursive estimate  $X_{k+1} = e^{-\alpha_k [X_k X_k^\top, N]} X_k$ , where*

$$\alpha_k = \frac{\|[X_k X_k^\top, N]\|^2}{2\sqrt{p} \|N[X_k X_k^\top, N]\|^2} \tag{13}$$

satisfies  $\Delta R_N(X_k, \alpha_k) = R_N(X_{k+1}) - R_N(X_k) > 0$ .

PROOF. Denote  $X_{k+1}(\tau) = e^{-\tau [X_k X_k^\top, N]} X_k$  for an arbitrary time-step  $\tau$ . Direct calculations show

$$\begin{aligned} \frac{d}{dt} \Delta R_N(X_k, \tau) &= -2 \text{tr}(X_{k+1}^\top(\tau) N [X_k X_k^\top, N] X_{k+1}(\tau)), \\ \frac{d}{dt} \Delta R_N(X_k, \tau) &= +4 \text{tr}(X_{k+1}^\top(\tau) N [X_k X_k^\top, N]^2 X_{k+1}(\tau)). \end{aligned}$$

Taylor’s formula for  $\Delta R_N(X_k, \tau)$  gives

$$\begin{aligned} \Delta R_N(X_k, \tau) &= -2\tau \text{tr}(X_k^\top N [X_k X_k^\top, N] X_k) \\ &\quad + 4\tau^2 \int_0^1 \text{tr}(X_{k+1}^\top(\tau) N [X_k X_k^\top, N]^2 X_{k+1}(\tau)) (1-s) ds \\ &\geq 2\tau \|[X_k X_k^\top, N]\|^2 \\ &\quad - 4\tau^2 \int_0^1 \|X_{k+1}(\tau) X_{k+1}^\top(\tau)\| \|N [X_k X_k^\top, N]\|^2 (1-s) ds \\ &= 2\tau \|[X_k X_k^\top, N]\|^2 - 2\tau^2 \sqrt{p} \|N [X_k X_k^\top, N]\|^2 =: \Delta R'_N(X_k, \tau). \end{aligned}$$

The quadratic nature of  $R'_N(X_k, \tau)$  yields a unique maximum occurring at  $\tau = \alpha_k$  given by (13). Observe that if  $\text{grad } R_N(X_k) \neq 0$ , then  $\|[X_k X_k^\top, N]\|^2 \neq 0$  and thus  $R'_N(X_k, \tau) > 0$ . The result follows since  $\Delta R_N(X_k, \tau) \geq \Delta R'_N(X_k, \tau) > 0$ .

**THEOREM 3.2.** *Let  $N = N^T$  be a real symmetric  $n \times n$  matrix, and  $p$  an integer with  $1 \leq p \leq n$ . Denote the eigenvalues of  $N$  by  $\lambda_1 \geq \dots \geq \lambda_n$ . For a given estimate  $X_k \in \text{St}(p, n)$ , let  $\alpha_k$  be given by (13). The Rayleigh gradient algorithm,*

$$X_{k+1} = e^{-\alpha_k [X_k X_k^T, N]} X_k,$$

*has the following properties.*

- (i) *The algorithm defines an iteration on  $\text{St}(p, n)$ .*
- (ii) *Fixed points of the algorithm are critical points of  $R_N$ ,  $X \in \text{St}(p, n)$  such that  $[X X^T, N] = 0$ . The columns of a fixed point of (11) form a basis for a  $p$ -dimensional eigenspace of  $N$ .*
- (iii) *If  $X_k$ , for  $k = 1, 2, \dots$ , is a solution to the algorithm, then the real sequence  $R_N(X_k)$  is strictly monotonic increasing unless there is some  $k \in \mathbb{N}$  with  $X_k$  a fixed point of the algorithm.*
- (iv) *Let  $X_k$ , for  $k = 1, 2, \dots$ , be a solution to the algorithm. Then  $X_k$  converges to a critical level set of  $R_N$  on  $\text{St}(p, n)$ .*
- (v) *All critical level sets of  $R_N$  are unstable except the set for which the Rayleigh quotient is maximised. The columns of an element of the maximal critical level set form a basis for the maximal eigenspace of  $N$ .*

**PROOF.** Part (i) follows from the observation that  $e^{-\alpha_k [X_k X_k^T, N]}$  is orthogonal. Part (ii) is a direct consequence of Lemma 3.1 (since  $\Delta R_N(X_k, \alpha_k) = 0$  if and only if  $X_k$  is a fixed point) and Theorem 2.1. Part (iii) also follows directly from Lemma 3.1.

To prove part (iv), observe that since  $\text{St}(p, n)$  is a compact set,  $R_N(X_k)$  is a bounded monotonically increasing sequence which must converge. As a consequence  $X_k$  converges to some level set of  $R_N$  such that for any  $X$  in this set  $\Delta R_N(X, \alpha(X)) = 0$ . Lemma 3.1 ensures that any  $X$  in this set is a fixed point of the recursion.

If  $X$  is a fixed point of the recursion whose columns do not span the maximal  $p$ -dimensional subspace of  $N$ , then it is clear that there exists an orthogonal matrix  $U \in O(n)$ , with  $\|U - I_n\|$  arbitrarily small and such that  $R_N(UX) > R_N(X)$ . As a consequence, the initial condition  $X_0 = UX$  ( $\|X_0 - X\|$  small) will give rise to a sequence of matrices  $X_k$  that diverges from the level set containing  $X$ , Lemma 3.1. This proves the first statement of (v), while the attractive nature of the remaining fixed points follows from La’Salle’s principle of invariance along with the Lyapunov function  $V(X) = (\sum_{i=1}^p \lambda_i) - R_N(X)$ .

**REMARK 3.3.** It is difficult to characterise the exact basin of attraction for the set of matrices whose columns span the maximal  $p$ -eigenspace of  $N$ . It is conjectured that the attractive basin for this set is all of  $\text{St}(p, n)$  except for other critical points.

REMARK 3.4. For a fixed initial condition  $X_0 \in \text{St}(p, n)$  let  $X_k$  be the solution to (11). Define  $H_k = X_k X_k^T$  and observe

$$H_{k+1} = e^{-\alpha_k[H_k, N]} H_k e^{\alpha_k[H_k, N]}. \tag{14}$$

Thus  $H_k$  can be written as a recursion on the set of symmetric rank  $p$  projection matrices  $\{H \in \mathbb{R}^{n \times n} \mid H = H^T, H^2 = H, \text{rank } H = p\}$ . The algorithm generated in this manner is known as the double-bracket algorithm [14], a discretization of the continuous-time double-bracket equation (10).

To illustrate the Rayleigh gradient algorithm a simulation has been included. A positive definite symmetric matrix  $N$  was randomly generated,

$$N = \begin{pmatrix} 13.3426 & 2.1180 & -3.8054 & -1.7068 \\ 2.1180 & 7.0640 & 0.3498 & 3.4856 \\ -3.8054 & 0.3498 & 5.1715 & 1.4446 \\ -1.7068 & 3.4856 & 1.4446 & 9.3422 \end{pmatrix},$$

and its maximal two-dimensional eigenspace (denoted  $D_2(N)$ ) was computed using a standard eigenvalue decomposition algorithm. A matrix  $U_2 \in \text{St}(4, 2)$  was chosen such that  $D_2(N) = \text{sp}(U_2)$  was the span of the columns of  $U_2$ . The Rayleigh gradient algorithm (11) was initialised with a randomly generated matrix  $X_0 \in \text{St}(4, 2)$  and run until the distance between the estimated and desired subspace was of order  $10^{-4}$ . The distance between  $\text{sp}(X_k)$  and the true maximal 2-dimensional eigenspace  $D_2(N)$  was computed by

$$\text{dist}(D_2(N), \text{sp}(X_k)) = \|X_k - U_2 U_2^T X_k\|^2 = p - \text{tr}(X_k^T U_2 U_2^T X_k),$$

the Frobenius norm of the projection of  $X_k$  onto the complement of the span of  $U_2$ . A plot of the distance versus iteration for this example is given by Figure 1. Plotting the distance on a logarithmic scale displays the linear convergence behaviour of the algorithm.

### 4. Computational considerations

In this section, two issues related to implementing (11) in a digital environment are discussed. Results in both the following subsections are aimed at reducing the computational cost associated with estimating the matrix exponential  $e^{-\alpha_k[X_k X_k^T, N]}$ , a transcendental  $n \times n$  matrix function. The result presented in Subsection 4.1 is also important in Section 5.

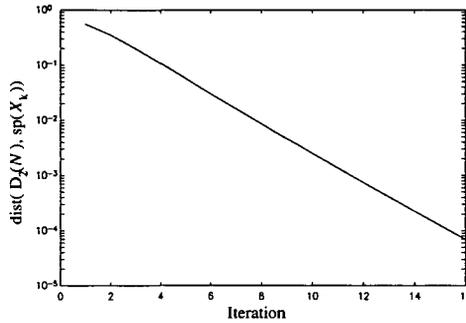


FIGURE 1. Plot of  $\text{dist}(D_2(N), \text{sp}(X_k))$  versus the iteration  $k$  for  $X_k$  a typical solution to (11).

**4.1. An equivalent formulation** To implement (11) on conventional computer architecture, the main computational cost for each step of the algorithm lies in computing the  $n \times n$  matrix exponential  $e^{-\alpha_k[X_k X_k^T, N]}$ . The following result provides an equivalent formulation of the algorithm which involves the related  $p \times p$  transcendental matrix functions “cos” and “sinc”. In many applications it is only required to compute the eigenspace associated with a couple of dominant eigenvalues,  $p \ll n$ , and considerable computational advantage can be obtained by using the following formulation.

Define the matrix function  $\text{sinc} : \mathbb{R}^{p \times p} \rightarrow \mathbb{R}^{p \times p}$  by the convergent infinite sum

$$\text{sinc}(A) = I_p - \frac{A^2}{3!} + \frac{A^4}{5!} - \frac{A^6}{7!} + \dots$$

Observe that  $A \text{sinc}(A) = \sin(A)$  and thus, if  $A$  is invertible,  $\text{sinc}(A) = A^{-1} \sin(A)$ . Define the matrix function  $\text{cos}(A)$  by an analogous power series expansion. The matrix functions  $\text{cos}$  and  $\text{sinc}$  are related by  $\text{cos}^2(A) = I_p - A^2 \text{sinc}^2(A)$ .

**LEMMA 4.1.** *Let  $N = N^T$  be a real symmetric  $n \times n$  matrix with eigenvalues  $\lambda_1 \geq \dots \geq \lambda_n$ , and  $\alpha_k$ , for  $k = 1, 2, \dots$ , be a sequence of real positive numbers. If  $X_0 \in \text{St}(p, n)$  is an initial condition that is not a critical point of  $R_N(X)$ , then*

$$\begin{aligned} X_{k+1} &= e^{-\alpha_k[X_k X_k^T, N]} X_k \\ &= X_k \left( \text{cos}(\alpha_k Y_k) - \alpha_k X_k^T N X_k \text{sinc}(\alpha_k Y_k) \right) + \alpha_k N X_k \text{sinc}(\alpha_k Y_k), \end{aligned} \tag{15}$$

where the power expansions for  $\text{cos}(\alpha_k Y_k)$  and  $\text{sinc}(\alpha_k Y_k)$  are determined by the positive semi-definite matrix  $Y_k^2 \in \mathbb{R}^{p \times p}$

$$Y_k^2 = X_k^T N (I_n - X_k X_k^T) N X_k = X_k^T N^2 X_k - (X_k^T N X_k)^2. \tag{16}$$

REMARK 4.2. The matrix  $Y_k$  need not be explicitly calculated as the power series expansions of sinc and cos depend only on  $Y_k^2$ .

PROOF. The proof follows from a power series expansion of  $e^{-\alpha_k[X_k X_k^T, N]} X_k$ ,

$$X_{k+1} = \left( \sum_{l=0}^{\infty} \frac{1}{l!} (-\alpha_k [X_k X_k^T, N])^l \right) \cdot X_k. \tag{17}$$

Simple algebraic manipulations lead to the relation

$$[X_k X_k^T, N]^2 X_k = -X_k Y_k^2, \tag{18}$$

where  $Y_k^2$  is defined by (16).

Using (18) it is possible to rewrite (17) as a power series in  $(-Y_k^2)$

$$X_{k+1} = \sum_{m=0}^{\infty} \left( \frac{(-\alpha_k)^{2m}}{(2m)!} X_k (-Y_k^2)^m + \frac{(-\alpha_k)^{2m+1}}{(2m+1)!} \left( X_k (X_k^T N X_k) - N X_k \right) (-Y_k^2)^m \right), \tag{19}$$

where the first and second terms in the summation follow from the odd and the even powers of  $[X_k X_k^T, N]^l X_k$  respectively. On rewriting this as two separate power series in  $(-Y_k^2)$

$$\begin{aligned} X_{k+1} &= X_k \sum_{m=0}^{\infty} \frac{(-\alpha_k)^{2m}}{(2m)!} (-Y_k^2)^m - \alpha_k \left( X_k (X_k^T N X_k) - N X_k \right) \sum_{m=0}^{\infty} \frac{(-\alpha_k)^{2m}}{(2m+1)!} (-Y_k^2)^m \\ &= X_k \cos(\alpha_k Y_k) - \alpha_k \left( X_k (X_k^T N X_k) - N X_k \right) \text{sinc}(\alpha_k Y_k), \end{aligned}$$

the result follows by rearranging terms.

**4.2. Padé approximations of the exponential** It is also of interest to consider approximate methods for calculating matrix exponentials. In particular, one is interested in methods that will not violate the constraint  $X_{k+1} \in \text{St}(p, n)$ . A standard approximation used for calculating the exponential function is a Padé approximation of order  $(n, m)$  where  $n > 0$  and  $m > 0$  are integers [10, page 557]. For example, a (1,1) Padé approximation of the exponential is

$$e^{-\alpha_k [X_k X_k^T, N]} = \left( I_n + \frac{\alpha_k}{2} [X_k X_k^T, N] \right)^{-1} \left( I_n - \frac{\alpha_k}{2} [X_k X_k^T, N] \right).$$

A key observation is that when  $n = m$  and the exponent is skew-symmetric, the resulting Padé approximate is orthogonal. Thus

$$X_{k+1} = \left( I_n + \frac{\alpha_k}{2} [X_k X_k^T, N] \right)^{-1} \left( I_n - \frac{\alpha_k}{2} [X_k X_k^T, N] \right) X_k, \tag{20}$$

with initial condition  $X_0 \in \text{St}(p, n)$ , defines an iteration on  $\text{St}(p, n)$  which approximates the Rayleigh gradient algorithm (11). Of course, in practice one would use an algorithm such as Gaussian elimination [10, page 92] to solve the linear system equations

$$\left(I_n + \frac{\alpha_k}{2}[X_k X_k^T, N]\right) X_{k+1} = \left(I_n - \frac{\alpha_k}{2}[X_k X_k^T, N]\right) X_k$$

for  $X_{k+1}$  rather than computing the inverse explicitly.

The algorithm defined by (20) can also be rewritten in a similar form to that obtained in Lemma 4.1. Consider the power series expansion

$$\left(I_n + \frac{\alpha_k}{2}[X_k X_k^T, N]\right)^{-1} = \sum_{i=0}^{\infty} \left(-\frac{\alpha_k}{2}[X_k X_k^T, N]\right)^i.$$

From here it is easily shown that

$$X_{k+1} = -X_k + \left(2X_k - \alpha_k(X_k(X_k^T N X_k) - N X_k)\right) \left(I_p + \frac{\alpha_k^2}{4} Y_k^2\right)^{-1}, \tag{21}$$

where  $Y_k^2 \in \mathbb{R}^{p \times p}$  is given by (16).

### 5. Comparison with classical algorithms

In this section the relationship between the Rayleigh gradient algorithm (11) and some classical algorithms for determining the maximal eigenspace of a symmetric matrix are investigated. A good discussion of the power method and the steepest ascent method for determining a single maximal eigenvalue of a symmetric matrix is given by Faddeev and Faddeeva [8]. Practical issues arising in implementing these algorithms along with direct generalizations to eigenspace methods are covered by Golub and Van Loan [10].

**5.1. The power method** In this subsection, the algorithm (11) in the case where  $p = 1$  is considered. It is shown that for a certain choice of time-step  $\alpha_k$ , the algorithm (11) is the classical power method.

In the case where  $p = 1$  then  $\text{St}(1, n) = \{x \in \mathbb{R}^n \mid \|x\| = 1\} = S^{n-1}$ , the  $(n - 1)$ -dimensional sphere in  $\mathbb{R}^n$ . The usual representation of the tangent space of  $S^{n-1}$  is  $T_x S^{n-1} = \{\xi \in \mathbb{R}^n \mid \xi^T x = 0\}$  and the Riemannian metric induced from the standard metric on  $\mathbb{R}^n$  is  $\langle\langle \xi, \eta \rangle\rangle = \xi^T \eta$ , for  $\xi, \eta$  in  $T_x S^{n-1}$  [11, page 25]. It is easily verified that these constructions are equivalent to considering  $\text{St}(1, n)$  as a homogeneous space.

**THEOREM 5.1.** *Let  $N = N^T$  be a real symmetric  $n \times n$  matrix with eigenvalues  $\lambda_1 \geq \dots \geq \lambda_n$ . For  $x_k \in S^{n-1}$  let  $\alpha_k$  be given by*

$$\alpha_k = \frac{y^2}{2\sqrt{2}\|N[x x^T, N]^2\|}, \tag{22}$$

where  $y_k \in \mathbb{R}$  is given by

$$y_k = \left( x_k^T N^2 x_k - (x_k^T N x_k)^2 \right)^{\frac{1}{2}}. \tag{23}$$

For  $x_0 \in St(1, n) = S^{n-1}$  an arbitrary initial condition, the following hold.

(i) The formula

$$x_{k+1} = e^{-\alpha_k [x_k x_k^T, N]} x_k$$

defines a recursive algorithm on  $S^{n-1}$ .

- (ii) Fixed points of the rank-1 Rayleigh gradient algorithm are the critical points of  $r_N$  on  $S^{n-1}$ , and are exactly the eigenvectors of  $N$ .
- (iii) If  $x_k$ , for  $k = 1, 2, \dots$  is a solution to the Rayleigh gradient algorithm, then the real sequence  $r_N(x_k)$  is strictly monotonic increasing, unless  $x_k$  is an eigenvector of  $N$ .
- (iv) For a given  $x_k \in S^{n-1}$  which is not an eigenvector of  $N$ , then  $y_k \neq 0$  and

$$x_{k+1} = \left( \cos(\alpha_k y_k) - x_k^T N x_k \frac{\sin(\alpha_k y_k)}{y_k} \right) x_k + \frac{\sin(\alpha_k y_k)}{y_k} N x_k. \tag{24}$$

- (v) Let  $x_k$ , for  $k = 1, 2, \dots$  be a solution to the rank-1 Rayleigh gradient algorithm. Then  $x_k$  converges to an eigenvector of  $N$ .
- (vi) All eigenvectors of  $N$ , considered as fixed points of (24), are unstable, except the eigenvector corresponding to the maximal eigenvalue  $\lambda_1$ , which is exponentially stable.

PROOF. Parts (i)–(iii) follow directly from Theorem 3.2. To see part (iv) observe that  $y_k = \|\text{grad } r_N(x_k)\|$  and  $y_k = 0$  if and only if  $\text{grad } r_N(x_k) = 0$  and  $x_k$  is an eigenvector of  $N$ . The recursive iteration (24) now follows directly from Lemma 4.1, with the substitution  $\text{sinc}(\alpha_k y_k) = \sin(\alpha_k y_k)/(\alpha_k y_k)$ . Parts (v) and (vi) again follow directly from Theorem 3.2.

REMARK 5.2. Equation (24) involves only  $N x_k$ ,  $x_k^T N x_k$  and  $(N x_k)^T (N x_k)$  vector computations. This structure is especially of interest when sparse or structured matrices  $N$  are considered.

A geodesic (or great circle) on  $S^{n-1}$ , passing through  $x$  at time  $t = 0$ , can be written

$$\gamma(t) = \cos(t)x - \sin(t)V, \tag{25}$$

where  $V = \dot{\gamma}(0)$  is a unit vector orthogonal to  $x$ . Choosing  $V(x_k) = \text{grad } r_N(x_k)/\|\text{grad } r_N(x_k)\|$ ,  $x = x_k$  and evaluating  $\gamma(t)$  at time  $t = \alpha_k \|\text{grad } r_N(x_k)\|$

gives (24). Thus, (24) is a geodesic interpolation of (8), the solution to the rank-1 Rayleigh gradient flow (7).

For a symmetric  $n \times n$  matrix  $N = N^T$  the classical power method is computed using the recursive formulae [10, page 351]

$$z_k = Nx_k \quad x_{k+1} = z_k / \|z_k\|. \tag{26}$$

The renormalisation operation is necessary if the algorithm is to be numerically stable. The following lemma shows that for  $N$  positive semi-definite and a particular choice of  $\alpha_k$  the rank-1 Rayleigh gradient algorithm (24) is exactly the power method (26).

LEMMA 5.3. Let  $N = N^T$  be a positive semi-definite  $n \times n$  matrix. For  $x_k \in S^{n-1}$  (not an eigenvector of  $N$ ) then  $\|\text{grad } r_N(x_k)\| \leq \|Nx_k\|$ . Let  $\alpha_k$  be given by

$$\alpha_k = \frac{1}{\|\text{grad } r_N(x_k)\|} \sin^{-1} \left( \frac{\|\text{grad } r_N(x_k)\|}{\|Nx_k\|} \right), \tag{27}$$

where  $\sin^{-1} \left( \frac{\|\text{grad } r_N(x_k)\|}{\|Nx_k\|} \right) \in (0, \pi/2)$ . Then

$$\frac{Nx_k}{\|Nx_k\|} = \left( \cos(\alpha_k y_k) - x_k^T Nx_k \frac{\sin(\alpha_k y_k)}{y_k} \right) x_k + \frac{\sin(\alpha_k y_k)}{y_k} Nx_k,$$

where  $y_k$  is given by (23).

PROOF. Observe that  $\|\text{grad } r_N(x_k)\|^2 = y_k^2 = \|Nx_k\|^2 - (x_k^T Nx_k)^2 \geq 0$  and thus  $\|\text{grad } r_N(x_k)\| \leq \|Nx_k\|$ . Consider the 2-dimensional linear subspace  $\text{sp}\{x_k, Nx_k\}$  of  $\mathbb{R}^n$ . The new estimate  $x_{k+1}$  generated using either (24) or (26) will lie in  $\text{sp}\{x_k, Nx_k\}$  (see Figure 2).

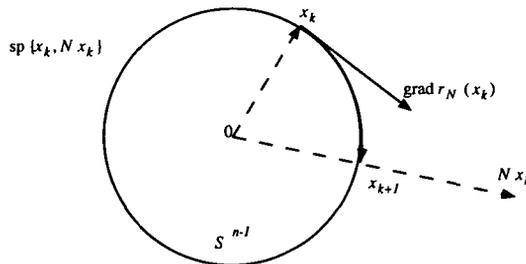


FIGURE 2. The geometric relationship between the power-method iterate and the iterate generated by (24).

Set

$$\frac{Nx_k}{\|Nx_k\|} = \left( \cos(\tau y_k) - x_k^\top Nx_k \frac{\sin(\tau y_k)}{y_k} \right) x_k + \frac{\sin(\tau y_k)}{y_k} Nx_k,$$

for  $\tau > 0$  and observe that  $x_k$  and  $Nx_k$  are linearly independent. Then

$$\cos(\tau y_k) - x_k^\top Nx_k \frac{\sin(\tau y_k)}{y_k} = 0 \quad \text{and} \quad \frac{\sin(\tau y_k)}{y_k} = \frac{1}{\|Nx_k\|}.$$

Since  $N > 0$  is positive definite, a real solution to the first relation exists for which  $\tau y_k \in (0, \pi/2)$ . The time-step value is now obtained by computing the smallest positive root of the second relation.

Choosing  $N > 0$  positive definite in Lemma 5.3 ensures that (24) and (26) converge ‘generically’ to the same eigenvector. Conversely, if  $N$  is symmetric with eigenvalues  $\lambda_1 > \dots > \lambda_n, 0 > \lambda_n$  and  $|\lambda_n| > |\lambda_i|$ , then the power method will converge to the eigenvector associated with  $\lambda_n$  while (24) (equipped with time-step (22)) will converge to the eigenvector associated with  $\lambda_1$ . Nevertheless, one may still choose  $\alpha_k$  using (27), with the inverse sine operation chosen to lie in the interval

$$\sin^{-1} \left( \frac{\|\text{grad } r_N(x_k)\|}{\|Nx_k\|} \right) \in (\pi/2, \pi),$$

such that (24) and (26) are equivalent. In this case the geodesics corresponding to each iteration of (24) are describing great circles travelling almost from pole to pole of the sphere.

**5.2. The steepest ascent algorithm** The *gradient ascent algorithm* for the Rayleigh quotient  $r_N$  is the recursion [8, page 430]

$$z_k = x_k + s_k \text{grad } r_N(x_k) \quad x_{k+1} = \frac{z_k}{\|z_k\|} \tag{28}$$

where  $s_k > 0$  is a real number termed the *step-size*, and  $\text{grad } r_N(x_k)$  is the Euclidean gradient computed in  $\mathbb{R}^n$ . It is easily verified that the  $k + 1$ -th iterate of (28) will also lie on the 2-dimensional linear subspace  $\text{sp}\{x_k, Nx_k\}$  of  $\mathbb{R}^n$ . Indeed, for  $x_k$  not an eigenvector of  $N$ , (24) and (28) are equivalent when

$$s_k = \frac{1}{y_k^2} \left( \frac{1}{\cos(\alpha_k y_k)} - 1 \right). \tag{29}$$

The optimal step-size for the steepest ascent algorithm (that is,  $r_N(x_{k+1}(s_k^{\text{opt}})) \geq r_N(x_{k+1}(s_k))$  for any  $s_k \in \mathbb{R}$ ) is [8, page 433]

$$s_k^{\text{opt}} = 2 \left( r_N(x_k) - r_N(\text{grad } r_N(x_k)) + \left\{ [r_N(x_k) - r_N(\text{grad } r_N(x_k))]^2 + 4\|r_N(x_k)\|^2 \right\}^{\frac{1}{2}} \right)^{-1}. \tag{30}$$

It follows directly that the optimal time-step selection for (24) is given by

$$\alpha_k = \frac{1}{y_k} \cos^{-1} \left( \frac{1}{1 + (s_k^{\text{opt}})^2 y_k^2} \right).$$

Substituting directly into (24) and analytically computing the composition of cos and sin with  $\cos^{-1}$  gives

$$x_{k+1} = \frac{1}{1 + (s_k^{\text{opt}})^2 y_k^2} \left[ \left( 1 - s_k^{\text{opt}} x_k^T N x_k \sqrt{2 + (s_k^{\text{opt}})^2 y_k^2} \right) x_k + \left( s_k^{\text{opt}} \sqrt{2 + (s_k^{\text{opt}})^2 y_k^2} \right) N x_k \right] \tag{31}$$

with  $s_k^{\text{opt}}$  given by (30). This recursion provides an optimal steepest ascent algorithm with scaling factor  $\frac{1}{1 + (s_k^{\text{opt}})^2 y_k^2}$ , which converges to one as  $x_k$  converges to an eigenvector of  $N$ .

**5.3. The generalised power method** In both the power method and the steepest ascent algorithm the rescaling operation preserves the computational stability of the calculation. To generalise classical methods to the case where  $p > 1$ , (that is,  $X_k \in \text{St}(p, n)$ ), one must decide on a procedure to renormalise new estimates to lie on  $\text{St}(p, n)$ . Thus a generalised power method may be written abstractly

$$Z_k = N X_k \quad X_{k+1} = \text{rescale}(Z_k). \tag{32}$$

Since the span of the columns of  $X_k$  (denoted  $\text{sp}(X_k)$ ) is the quantity in which one is interested, the rescaling operation is usually computed by generating an orthonormal basis for  $\text{sp}(Z_k)$  (that is, using the Gram-Schmidt algorithm [10, page 218]). Thus  $X_{k+1} = Z_k G$ , and  $X_{k+1}^T X_{k+1} = I_p$ , where  $G \in \mathbb{R}^{p \times p}$  contains the coefficients which orthonormalise the columns of  $Z_k$ . When  $Z_k$  is full rank then  $G$  is invertible and the factorization  $Z_k = X_{k+1} G^{-1}$  can be computed as a  $QR$  factorisation of  $Z_k$  [10, page 211]. The matrix  $G$  depends on the particular algorithm employed in computing an orthonormal basis for  $Z_k$ .

When  $N > 0$  is positive definite, the power method will act to maximise the generalised Rayleigh quotient  $R_N$  (5). Different choices of  $G$  in the rescaling operation, however, will affect the performance of the power method with respect to the relative change in  $R_N$  at each iteration. The optimal choice of  $G$  (for maximising the increase in Rayleigh quotient) for the  $k$ -th step of (32) is given by a solution of the optimization problem

$$\max_{\{G \in \mathbb{R}^{p \times p} \mid G^T Z_k^T Z_k G = I_p\}} \text{tr} \left( G^T Z_k^T N Z_k G \right),$$

where  $Z_k = N X_k$ . The cost criterion  $\text{tr} \left( G^T Z_k^T N Z_k G \right) = R_{Z_k^T N Z_k} (G)$  is a Rayleigh quotient while the constraint set is similar in structure to  $\text{St}(p, n)$ . Indeed, it appears

that this optimization problem is qualitatively the same as explicitly solving for the principal components of  $N$ .

One may still hope to obtain a similar result to Lemma 5.3 relating the generalised power method to the Rayleigh gradient algorithm (11). Unfortunately, this not the case except in nongeneric cases.

LEMMA 5.4. *Let  $N = N^T$  be a symmetric  $n \times n$  matrix. For any  $X_k \in \text{St}(p, n)$  let  $Y_k$  be the unique symmetric, positive semi-definite square root of  $Y_k^2 = X_k^T N^2 X_k - (X_k^T N X_k)^2$ . There exists a matrix  $G \in \mathbb{R}^{p \times p}$  and scalar  $\alpha_k > 0$  such that*

$$N X_k G = e^{-\alpha_k [X_k X_k^T, N]} X_k, \tag{33}$$

if and only if one can solve

$$\sin^2(\alpha_k Y_k) X_k^T N X_k = \cos(\alpha_k Y_k) \sin(\alpha_k Y_k) Y_k \tag{34}$$

for  $\alpha_k$ .

PROOF. Assume that there exists a matrix  $G$  and a scalar  $\alpha_k > 0$  such that (33) holds. Observe that  $\text{rank}(e^{-\alpha_k [X_k X_k^T, N]} X_k) = p$  and thus  $\text{rank}(N X_k) = p$ . Similarly  $G \in \mathbb{R}^{p \times p}$  is non-singular.

Premultiplication (33) by  $G^T X_k^T N$  and use of the constraint relation  $G^T X_k^T N^2 X_k G = I_p$  gives

$$I_p = G^T X_k^T N e^{-\alpha_k [X_k X_k^T, N]} X_k.$$

Since one need only consider the case where  $G$  is invertible, it follows that

$$G^{-1} = X_k^T e^{\alpha_k [X_k X_k^T, N]} N X_k.$$

Lengthy matrix manipulations yield

$$X_k^T [X_k X_k^T, N]^{2l} N X_k = (-1)^l Y_k^{2l} X_k^T N X_k, \quad \text{for } l = 0, 1, \dots,$$

and

$$X_k^T [X_k X_k^T, N]^{2l+1} N X_k = (-1)^l Y_k^{2l+2} \quad \text{for } l = 0, 1, \dots$$

On expanding  $e^{\alpha_k [X_k X_k^T, N]}$  as a power series in  $Y_k^2$  and then grouping terms suitably (see Subsection 4.1), one obtains

$$G^{-1} = \cos(\alpha_k Y_k) X_k^T N X_k + \sin(\alpha_k Y_k) Y_k.$$

If we use (15) for  $e^{-\alpha_k [X_k X_k^T, N]} X_k$  then (33) becomes

$$\begin{aligned} N X_k &= e^{-\alpha_k [X_k X_k^T, N]} X_k G^{-1} \\ &= \left( X_k \cos(\alpha Y_k) - \alpha_k [X_k X_k^T, N] X_k \text{sinc}(\alpha_k Y_k) \right) \left( \cos(\alpha_k Y_k) X_k^T N X_k + \sin(\alpha_k Y_k) Y_k \right). \end{aligned}$$

Premultiplying this by  $X_k^T$  yields

$$X_k^T N X_k = \cos^2(\alpha_k Y_k) X_k^T N X_k^T + \cos(\alpha_k Y_k) \sin(\alpha_k Y_k) Y_k,$$

and thus

$$\sin^2(\alpha_k Y_k) X_k^T N X_k^T = \cos(\alpha_k Y_k) \sin(\alpha_k Y_k) Y_k.$$

This shows that (33) implies (34). If  $\alpha_k$  solves (34) then defining  $G^{-1} = X_k^T e^{\alpha_k [X_k X_k^T, N]} N X_k$  ensures (33) also holds which completes the proof.

Write  $Y_k = \sum_{i=1}^p \beta_i y_i y_i^T$ , where  $\{y_1, \dots, y_p\}$  is a set of orthonormal eigenvectors for  $Y_k$ , whose eigenvalues are denoted  $\beta_i \geq 0$  for  $i = 1, \dots, p$ . Then (34) becomes

$$\sum_{i=1}^p \sin^2(\alpha_k \beta_i) y_i y_i^T X_k^T N X_k = \sum_{i=1}^p \beta_i \cos(\alpha_k \beta_i) \sin(\alpha_k \beta_i) y_i y_i^T.$$

Fixing  $i$  and premultiplying by  $y_i^T$  while also postmultiplying by  $y_i$  gives the following  $p$  equations for  $\alpha_k$ :

$$\text{either } \sin(\alpha_k \beta_i) = 0 \text{ or } \cot(\alpha_k \beta_i) = \frac{1}{\beta_i} y_i^T X_k^T N X_k y_i,$$

for  $i = 1, \dots, p$ . It follows that either from the first relation  $\alpha_k \beta_i = m\pi$  for some integer  $m$  or from the second relation, that

$$\cot(\alpha_k) = \frac{\beta_i + \cot(\beta_i) y_i^T X_k^T N X_k y_i}{\beta_i \cot(\beta_i) + y_i^T X_k^T N X_k y_i}$$

for each  $i = 1, \dots, p$ . One can easily confirm from this that the  $p$  equations will fail to have a consistent solution for arbitrary choices of  $X_k$  and  $N$ . Thus, generically, the Rayleigh gradient algorithm (11) does not correspond to the generalised power method (32) for any choice of rescaling operation or time-step selection.

### Acknowledgement

The authors wish to acknowledge the funding of the activities of the Cooperative Research Centre for Robust and Adaptive Systems by the Australian Commonwealth Government under the Cooperative Research Centres Program, and separate support by Boeing Commercial Aircraft Cooperation Inc.

## References

- [1] G. Ammar and C. Martin, "The geometry of matrix eigenvalue methods", *Acta Applicandae Mathematicae* **5** (1986) 239–278.
- [2] A. M. Bloch, "A completely integrable Hamiltonian system associated with line fitting in complex vector spaces", *Bull. Amer. Math. Soc.* **12** (1985) 250–254.
- [3] R. W. Brockett, *Differential geometry and the design of gradient algorithms*, Vol. 54 *Proc. Symp. Pure Mathematics* (1993) 69–92.
- [4] C. I. Byrnes and J. C. Willems, "Least-squares estimation, linear programming and momentum: A geometric parametrization of local minima", *IMA J. Math. Control and Information* **3** (1986) 103–118.
- [5] M. T. Chu, "On the continuous realization of iterative processes", *SIAM Review* **30** (1988) 375–387.
- [6] P. Deift, T. Nanda and C. Tomei, "Ordinary differential equations for the symmetric eigenvalue problem", *SIAM J. of Numer. Anal.* **20** (1983) 1–22.
- [7] J. J. Duistermaat, J. A. C. Kolk and V. S. Varadarajan, "Functions, flows and oscillatory integrals on flag manifolds and conjugacy classes in real semi-simple Lie groups", *Composito Mathematica* **49** (1983) 309–398.
- [8] D. K. Faddeev and V. N. Faddeeva, *Computational methods of linear algebra* (W. H. Freeman and Co., San Francisco, 1963).
- [9] C. G. Gibson, *Singular points of smooth mappings*, Vol. 25 *Res. Notes in Math.* (Pitman, London, United Kingdom, 1979).
- [10] G. H. Golub and C. F. Van Loan, *Matrix computations* (The Johns Hopkins University Press, Baltimore, Maryland U.S.A., 1989).
- [11] U. Helmke and J. B. Moore, *Optimization and dynamical systems*, Communications and Control Engineering (Springer-Verlag, London, 1994).
- [12] R. A. Horn and C. R. Johnson, *Matrix analysis* (Cambridge University Press, Cambridge, U.K., 1985).
- [13] R. E. Mahony, "Optimization algorithms on homogeneous spaces: with applications in linear systems theory", Ph. D. Thesis, Department of Systems Engineering, Canberra, Australia, 1994.
- [14] J. B. Moore, R. E. Mahony and U. Helmke, "Numerical gradient algorithms for eigenvalue and singular value calculations", *SIAM J. Matrix Analysis* (1994).
- [15] E. Oja, "A simplified neuron model as a principal component analyzer", *J. Math. Biology* **15** (1982) 267–273.
- [16] E. Oja, "Neural networks, principal components, and subspaces", *Intern. J. Neural Systems* **1** (1989) 61–68.
- [17] F. W. Warner, *Foundations of differentiable manifolds and Lie groups*, Graduate texts in Mathematics (Springer-Verlag, New York, U.S.A., 1983).
- [18] W.-Y. Yan, U. Helmke and J. B. Moore, "Global analysis of Oja's flow for neural networks", *IEEE Tran. Neural Networks* **5** (1994), 674–683.