REPORTS
IN
INFORMATICS

ISSN 0333-3590

A low complexity Lie group method on the Stiefel manifold

Stein Krogstad

REPORT NO **July 2001**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL `http://www.ii.uib.no/publikasjoner/texrap/ps/2001-.ps`

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at `http://www.ii.uib.no/publikasjoner/texrap/`.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

A low complexity Lie group method on the Stiefel manifold.

Stein Krogstad*

Department of Informatics
University of Bergen
N-5020 Bergen
Norway

23rd July 2001

Abstract

A low complexity Lie group method for numerical integration of ordinary differential equations on the orthogonal Stiefel manifold is presented. Based on the quotient space representation of the Stiefel manifold we provide a representation of the tangent space suitable for Lie group methods. According to this representation a special type of generalized polar coordinates (GPC) is defined and used as a coordinate map. The GPC maps, recently proposed by Munthe-Kaas and Zanna, prove to adapt well to the Stiefel manifold. For the $n \times k$ matrix representation of the Stiefel manifold the arithmetic complexity of the method presented is of order nk^2, and for $n \gg k$ this leads to huge savings in computation time compared to ordinary Lie group methods. Numerical experiments compare the method to a standard Lie group method using the matrix exponential, and conclude that on the examples presented, the methods perform equally on both accuracy and maintaining orthogonality.

1 Introduction

Let $V_{n,k}$ denote the (real) Stiefel manifold which may be represented as the set of $n \times k$ matrices Q with orthonormal columns, i.e. $Q^T Q = I_k$. Among the most usual applications of numerical integration of equations evolving on the Stiefel manifold is the computation of Lyapunov exponents (see [5]), and an example is given in section 5.

We denote the *origin* in $V_{n,k}$ by $I_{n,k}$ which is the $n \times k$ identity matrix. Let $c(t)$ be a curve on $V_{n,k}$ such that $c(0) = I_{n,k}$ and $\dot{c}(0) = \Delta$. By differentiating the relation $c(t)^T c(t) = I_{n,k}$, we find that the tangent space at the origin consists of the $n \times k$ matrices

$$\Delta = \begin{pmatrix} A \\ B \end{pmatrix} \tag{1}$$

where A is $k \times k$ skew symmetric, and B is $(n-k) \times k$ arbitrary. Let $F \in \mathbb{R}^{n \times k}$. Obviously there are several ways to project F onto the space consisting of matrices of the form (1). Leaving the lower $(n-k) \times k$ block of F intact, we can choose any projection of the upper $k \times k$ block onto the space of skew symmetric matrices. The natural choice is to apply skew, i.e. $\text{skew}(M) = \frac{1}{2}(M - M^T)$, which leads to an orthogonal projector with respect to the Euclidian inner product. As we will see in section 2, this simple observation carries over to the tangent space at any point on the manifold.

Given an ordinary differential equation evolving on the Stiefel manifold

$$\dot{Q}(t) = F(Q, t) \quad Q(0) = Q_0 \tag{2}$$

*Email: stein@ii.uib.no

For many numerical methods it is convenient to reformulate the equation (2) to one of the form

$$\dot{Q}(t) = H(Q,t)Q(t) \quad Q(0) = Q_0 \tag{3}$$

where H is skew symmetric. There seems to be many different ways to formulate the equation (3), all being mathematically equivalent but not necessarily numerically equivalent, i.e they may lead to different numerical solutions of the equation. Examples of such formulations are given in [1] and [2]. In section 2 we formulate equation (3) inspired by [13] and [6], such that it fits the coordinate map defined in section 3. The results have deeper foundations in the theory of principal fibre bundles (see [9]). Given an equation of the form (3), a Lie group method is straight forward applicable. Never the less, a standard Lie group method such as the RKMK using the matrix exponential will include operations on the matrix H forcing an arithmetic complexity of order n^3 which is not satisfactory for $n \gg k$. We deal with this problem using generalized polar coordinates (GPC) defined by Munthe-Kaas and Zanna in [12]. The GPC maps based on the generalized polar decomposition of the Lie algebra (see [11] and [14]), seem to be ideal for numerical integration on the Stiefel manifold, and we are able to carry out the computation with complexity $O(nk^2)$. We remark that we do not count the operations of evaluating the the vectorfield F in (2). As in [2] we regard this as a *black box* operation, and thus it may be of order n^2k. In section 5 we present numerical experiments with this map showing huge savings compared to the matrix exponential, but which shows equal qualitative behavior.

2 The Stiefel Manifold and its tangent space

In preparing the tangent space of the Stiefel manifold for a Lie group method, the $n \times k$ matrix representation of the tangent space is, at least for starters, not convenient. In addition to the $n \times k$ representation of the Stiefel manifold, it is also common to represent it as the quotient space $O(n)/O(n-k)$ (see [13] or [6]). Here each point on the Stiefel manifold corresponds to an equivalence class $[\widetilde{Q}]$ consisting of $n \times n$ orthogonal matrices shearing the property that their first k columns coincide. We say that a $\widetilde{Q} \in O(n)$ is a *coset representative* of $Q \in V_{n,k}$ (in the $n \times k$ representation) whenever $Q = \widetilde{Q}I_{n,k}$. Following [13] we define $H_Q \subset O(n)$ to be the isotropy group at Q, i.e. the group consisting of all $\widetilde{Q} \in O(n)$ such that $\widetilde{Q}Q = Q$. It follows easily that

$$H_{I_{n,k}} = \left\{ \begin{pmatrix} I_k & 0 \\ 0 & C \end{pmatrix} : C \in O(n-k) \right\}$$

and that $H_Q = \widetilde{Q}H_{I_{n,k}}\widetilde{Q}^{-1}$, where $\widetilde{Q}$ is any coset representative for Q. Now for each Q, H_Q is a Lie subgroup of $O(n)$, and the Stiefel manifold can be represented as the quotient space $O(n)/H_Q$. We denote $\mathfrak{so}(n)$ the Lie algebra of $O(n)$, i.e. the space of $n \times n$ skew symmetric matrices. Letting $\mathfrak{h}_Q$ denote the Lie algebra of H_Q, one obtains a splitting of $\mathfrak{so}(n)$ into two subspaces:

$$\mathfrak{so}(n) = \mathfrak{m}_Q + \mathfrak{h}_Q \tag{4}$$

where the Lie subalgebra $\mathfrak{h}_Q$ may be represented by

$$\mathfrak{h}_Q = \left\{ \widetilde{Q} \begin{pmatrix} 0 & 0 \\ 0 & C \end{pmatrix} \widetilde{Q}^{-1} : C \in \mathfrak{so}(n-k) \right\} \tag{5}$$

and the space $\mathfrak{m}_Q$ by the matrices

$$\mathfrak{m}_Q = \left\{ \widetilde{Q} \begin{pmatrix} A & -B^T \\ B & 0 \end{pmatrix} \widetilde{Q}^{-1} : A \in \mathfrak{so}(k), B \in \mathbb{R}^{(n-k)\times k} \right\} \tag{6}$$

Thus any tangent vector at Q has the representation MQ where $M \in \mathfrak{m}_Q$, and $HQ = 0$ for any $H \in \mathfrak{h}_Q$. Also we emphasize that since $\mathfrak{h}_Q$ is the Lie algebra of the isotropy group at Q, we have $\exp(H)Q = Q$.

2.1 Projections onto the tangent space

Given any $F \in \mathbb{R}^{n\times k}$ there are several ways to define projections onto $T_{I_{n,k}}V_{n,k}$, i.e. the space containing matrices of the form (1). The obvious choice is take the skew part of the upper $k \times k$ block of F, and leaving the lower part intact. This projection is orthogonal with respect to the Euclidian inner product. At an arbitrary point Q, any tangent vector F has the representation $F = ZQ$ where Z is $n \times n$ skew symmetric. We wish to find a projection of Z onto the space $\mathfrak{m}_Q$, or more generally a projection $\Pi_{\mathfrak{m}_Q} : \mathbb{R}^{n\times n} \to \mathfrak{m}_Q$. Denoting $\mathfrak{m} = \mathfrak{m}_{I_{n,k}}$, we see from (6) that given $\Pi_{\mathfrak{m}}$, the corresponding $\Pi_{\mathfrak{m}_Q}$ is obtained by

$$\Pi_{\mathfrak{m}_Q} = \mathrm{Ad}_{\widetilde{Q}} \circ \Pi_{\mathfrak{m}} \circ \mathrm{Ad}_{\widetilde{Q}^{-1}} \tag{7}$$

where $\mathrm{Ad}_{\widetilde{Q}}(M) = \widetilde{Q}M\widetilde{Q}^{-1}$ and $\widetilde{Q}$ is any coset representative for Q. Letting π_{skew} be any projection onto the space of $k \times k$ skew symmetric matrices, we wish the operator $\Pi_{\mathfrak{m}}$ to work as follows:

$$\Pi_{\mathfrak{m}} : \begin{pmatrix} A & * \\ B & * \end{pmatrix} \mapsto \begin{pmatrix} \pi_{\text{skew}}(A) & -B^T \\ B & 0 \end{pmatrix}$$

where $*$ represents arbitrary blocks. By some manipulation we find that $\Pi_{\mathfrak{m}}$ working on a $Z \in \mathbb{R}^{n\times n}$ can be represented by:

$$\Pi_{\mathfrak{m}}(Z) = \beta(I_{n,k})I_{n,k}^T - I_{n,k}\beta(I_{n,k})^T + I_{n,k}\alpha(I_{n,k})I_{n,k}^T \tag{8}$$

where $\beta(I_{n,k}) = (I - I_{n,k}I_{n,k}^T)ZI_{n,k}$ and $\alpha(I_{n,k}) = \pi_{\text{skew}}\left(I_{n,k}^T Z I_{n,k}\right)$. Applying (7) to the expression above (recalling that $Q = \widetilde{Q}I_{n,k}$) we obtain

$$\Pi_{\mathfrak{m}_Q}(Z) = \beta(Q)Q^T - Q\beta(Q)^T + Q\alpha(Q)Q \tag{9}$$

where $\beta(Q) = (I - QQ^T)ZQ$ and $\alpha(Q) = \pi_{\text{skew}}(Q^T ZQ)$. If Z is all ready skew, then $Q^T ZQ$ also is skew and we simply have $\alpha(Q) = Q^T ZQ$.

It is worth noting that we do not need the matrix Z explicitly for calculating the projection. Regarding $F = ZQ$ as a *black box* operation, we find $\Pi_{\mathfrak{m}_Q}(Z)$ as above with $\beta(Q) = F - QQ^T F$ and $\alpha(Q) = \pi_{\text{skew}}(Q^T F)$.

2.2 Restricting ODEs to the Stiefel manifold

Given any linear system

$$\dot{Y}(t) = A(t)Y(t) = F(Y,t) \quad Y(0) = Y_0$$

where $A : \mathbb{R}^+ \to \mathbb{R}^{n\times n}$ and $Y : \mathbb{R}^+ \to \mathbb{R}^{n\times k}$. Restricting the system above to the Stiefel manifold is now done by projecting the right hand side of the above equation onto the tangent space of the Stiefel manifold to obtain

$$\dot{Q}(t) = \Pi_{\mathfrak{m}_Q}(A(t))\,Q(t) \quad Q(0) = Q_0$$

where Q_0 is some orthonormal factor of Y_0. This system has the desired *strong* skew symmetry property since obviously $\mathfrak{m}_Q \subset \mathfrak{so}(n)$. The importance of strong versus weak skew symmetry is discussed in [1].

Continuous QR decomposition: Consider the linear system

$$\dot{Y}(t) = A(t)Y(t)$$

where $Y \in \mathbb{R}^{n\times k}$ with $Y(0) = Y_0 = Q_0R_0$. If Y has full rank, there exists a continuous QR factorization of Y such that $Y(t) = Q(t)R(t)$. The differential equation for Q can be found by differentiating $Y = QR$, and can be expressed (see [5])

$$\dot{Q}(t) = A(t)Q(t) - QT \quad T_{ij} = \begin{cases} (Q^TAQ)_{ij} + (Q^TAQ)_{ji} & i < j \\ (Q^TAQ)_{ii} & i = j \\ 0 & i > j \end{cases}$$

Applying the the operator $\Pi_{\mathfrak{m}_Q}$ we reformulate the equation to

$$\dot{Q}(t) = \left(\beta(Q)Q^T - Q\beta(Q)^T + Q\alpha(Q)Q^T\right)Q(t)$$

where as above $\beta(Q) = (I - QQ^T)AQ$, and the α-part becomes $\alpha(Q) = \mathrm{tril}(Q^TAQ) - \mathrm{tril}(Q^TAQ)^T$. Here $\mathrm{tril}(M)$ denotes the strictly lower triangular part of M.

3 Generalized polar coordinates on the Stiefel manifold

In the original RKMK method [10], the matrix exponential is used as a coordinate map from the given Lie algebra to its corresponding Lie group. Other choices of coordinate maps are possible, and a comparison of different choices are discussed in [7]. In [12] the authors propose to use approximations to the exponential map by decomposing matrices with aid of projections resulting from inner automorphisms. There are mainly two good reasons for using these approximations as coordinate maps. Primarily, the approximations reside in the Lie group which is not the case for arbitrary approximations. Secondly the computation can be carried out with low complexity.

A matrix S is called *involutive* if $S^2 = I$. With respect to S we define the projection matrices

$$\Pi_S^{\pm} = \frac{1}{2}(I \pm S)$$

Also with respect to S we define the *inner automorphism* σ, mapping a matrix Lie algebra $\mathfrak{g}$ to itself, by $\sigma(Z) = SZS$. Corresponding to such a σ we now define the projection operators

$$\Pi_\sigma^{\pm} = \frac{1}{2}(I \pm \sigma)$$

Denote $P = \Pi_\sigma^-(Z)$ and $K = \Pi_\sigma^+(Z)$ for $Z \in \mathfrak{g}$. The projections decompose $\mathfrak{g}$ into two spaces where K belongs to a Lie subalgebra $\mathfrak{k} \subset \mathfrak{g}$, and P belongs to a *Lie triple system* $\mathfrak{p} \subset \mathfrak{g}$ (not a Lie algebra) closed under double commutators. The spaces $\mathfrak{k}$ and $\mathfrak{p}$ have the following "$\pm$" properties when they interact via commutators

$$[\mathfrak{k}, \mathfrak{k}] \subset \mathfrak{k} \qquad [\mathfrak{k}, \mathfrak{p}] \subset \mathfrak{p} \qquad [\mathfrak{p}, \mathfrak{p}] \subset \mathfrak{k} \tag{10}$$

In [12] the definition of generalized polar coordinates (GPC) is presented as follows:

- The exponential map $\exp : \mathfrak{g} \to G$ is a GPC on G.
- Let σ be an involutive automorphism on $\mathfrak{g}$. If $\tilde{\Phi} : \Pi_\sigma^+\mathfrak{g} \to G^\sigma$ is a GPC on G^σ , then a map $\Phi : \mathfrak{g} \to G$ defined as
$$\Phi(Z) = \exp(\Pi_\sigma^- Z) \cdot \tilde{\Phi}(\Pi_\sigma^+ Z) \tag{11}$$
is a GPC on G.

Here G^σ denotes the Lie subgroup of G containing the image of $\tilde{\Phi}$. Given a GPC Φ, for any element $g \in G$ sufficiently close to the identity, there exists a $Z \in \mathfrak{g}$ such that $g = \Phi(Z)$.

Let $P = \Pi_\sigma^- Z$ and $K = \Pi_\sigma^+ Z$. It is shown in [12] that the computation of $\phi(P)$ for any analytic function ϕ, can be obtained according to the formula

$$\phi(P) = \phi(0)I + \phi_1(\Theta)P + P\phi_1(\Theta) + P\phi_2(\Theta)P + \phi_2(\Theta)\Theta \tag{12}$$

where $\Theta = P^2\Pi_S^-$, and

$$\phi_1(x) = \frac{1}{2\sqrt{x}}\left(\phi(\sqrt{x}) - \phi(-\sqrt{x})\right) \tag{13}$$

$$\phi_2(x) = \frac{1}{2x}\left(\phi(\sqrt{x}) + \phi(-\sqrt{x}) - 2\phi(0)\right) \tag{14}$$

3.1 The coordinate map

Let Z be any skew symmetric matrix and let $Q \in V_{n,k}$. By choosing the involutive $S = I - 2QQ^T$, and the corresponding σ such that $\sigma(Z) = SZS$, we obtain the decomposition $Z = P + K$ where

$$P = \beta Q^T - Q\beta^T \tag{15}$$
$$K = Q\alpha Q^T + H \tag{16}$$

where α and β are short for $\alpha(Q) = Q^T ZQ$ and $\beta(Q) = (I - QQ^T)ZQ$ respectively, and H is an element in $\mathfrak{h}_Q$ (i.e. $HQ = 0$). Thus, the coordinate map we wish to use is

$$\Phi(Z) = \exp(P) \exp(K)$$

We note that the notation $\Phi(Z)$ is inaccurate since Φ also depends on our current whereabouts on the manifold. A better notation would be $\Phi_Q(Z)$, but we omit the Q to avoid cluttering the notation later on. By definition of the space $\mathfrak{h}_Q$ the two summands of K commute and we have

$$\begin{aligned} \exp(K)Q &= \exp(Q\alpha Q^T) \exp(H)Q \\ &= Q \exp(\alpha) \end{aligned} \tag{17}$$

We find a nice representation for $\exp(P)Q$ using the formula (12). Alternatively we can just use the relation $\beta^T Q = 0$ to obtain

$$\begin{aligned} P^{2k}Q &= Q\left(-\beta^T\beta\right)^k \\ P^{2k+1}Q &= \beta\left(-\beta^T\beta\right)^k \end{aligned}$$

By comparing these terms with $\exp(P) = \sum_{k=0}^{\infty} P^k/k!$, we obtain

$$\exp(P)Q = Q\phi_1\left(-\beta^T\beta\right) + \beta\phi_2\left(-\beta^T\beta\right)$$

where

$$\begin{aligned} \phi_1(x) &= \cos(\sqrt{-x}) \\ \phi_2(x) &= \begin{cases} 1 & \text{if } x = 0 \\ \frac{\sin(\sqrt{-x})}{\sqrt{-x}} & \text{if } x \neq 0 \end{cases} \end{aligned}$$

Summarizing we have the following expression for $\Phi(Z)$ acting on Q

$$\begin{aligned} \Phi(Z)Q &= \exp(P)\exp(K)Q \\ &= \left(Q\phi_1(-\beta^T\beta) + \beta\phi_2(-\beta^T\beta)\right)\exp(\alpha) \end{aligned} \tag{18}$$

Thus we have reduced an expression involving exponentiation of $n \times n$ matrices to an expression with analytic functions of $k \times k$ matrices. We note that as in (9), we do not need Z explicitly to calculate $\Phi(Z)Q$. We only need an $F \in T_Q V_{n,k}$ such that $F = ZQ$. In this way $\mathcal{R}_Q(F) := \Phi(Z)Q : T_Q V_{n,k} \to V_{n,k}$ is an example of a retraction as described in [2]. The numerical accuracy of the formula (18) relies heavily on the condition $Q^T\beta(Q) = 0$, but this should not cause problems if implemented properly.

3.2 The tangent map

Given a coordinate map $\Phi : \mathfrak{g} \to G$, the *right trivialized tangent* of Φ is defined as the mapping $\mathrm{d}\Phi : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ satisfying

$$\frac{d}{dt}\Phi(A(t)) = \mathrm{d}\Phi_{A(t)}(A'(t))\,\Phi(A(t))$$

Let as before $Z = P + K$, and let $\delta Z \in \mathfrak{g}$ be decomposed similarly as $\delta Z = \delta P + \delta K$. For our particular choice of coordinate map Φ we have (following [12])

$$\begin{aligned} \mathrm{d}\Phi_Z(\delta Z)\Phi(Z) &= \left.\frac{d}{dt}\right|_{t=0} \exp(P + t\delta P)\exp(K + t\delta K) \\ &= \mathrm{dexp}_P(\delta P)\exp(P)\exp(K) + \exp(P)\mathrm{dexp}_K(\delta K)\exp(K) \end{aligned} \tag{19}$$

which implies

$$\begin{aligned} \mathrm{d}\Phi_Z &= \mathrm{dexp}_P \Pi_\sigma^- + \mathrm{Ad}_{\exp(P)}\mathrm{dexp}_K \Pi_\sigma^+ \\ &= \frac{\exp(u) - 1}{u}\Pi_\sigma^- + \exp(u)\mathrm{dexp}_K \Pi_\sigma^+ \end{aligned}$$

where $u = \mathrm{ad}_P$. We used here the relation $\mathrm{Ad}_{\exp(P)} = \exp(\mathrm{ad}_P)$. Applying an analytic function to ad_P should be understood as inserting powers of ad_P into the functions Taylor expansion, where $\mathrm{ad}_P^0(\delta P) = \delta P$ and $\mathrm{ad}_P^k(\delta P) = [P, [P, \cdots [P, \delta P] \cdots]]$ (k brackets). The expression for $\mathrm{d}\Phi_Z$ can now be splitted into the spaces $\mathfrak{k}$ and $\mathfrak{p}$ according to (10). We omit the technical details here and present the formula for $\mathrm{d}\Phi_Z^{-1}$ analogous to [12]:

$$\mathrm{d}\Phi_Z^{-1} = \left(\Pi_\sigma^- + \mathrm{dexp}_K^{-1}\Pi_\sigma^+\right)\left(I + \mathrm{ad}_P\left(\theta_1\left(\mathrm{ad}_P^2\Pi_\sigma^-\right) - \Pi_\sigma^+\right) + \theta_2\left(\mathrm{ad}_P^2\Pi_\sigma^-\right)\right) \tag{20}$$

where

$$\theta_1(x) = \begin{cases} -\frac{1}{2} & \text{if } x = 0 \\ -2\frac{\sin^2(\sqrt{-x}/2)}{\sqrt{-x}\sin(\sqrt{-x})} & \text{if } x \neq 0 \end{cases} \tag{21}$$

$$\theta_2(x) = \begin{cases} 0 & \text{if } x = 0 \\ \frac{\sqrt{-x}}{\tan(\sqrt{-x})} - 1 & \text{if } x \neq 0 \end{cases} \tag{22}$$

For a better overview of the formula (20), we split it corresponding to Π_σ^- and Π_σ^+. When applying the tangent map to an $\delta Z = \delta P + \delta K$, we get :

$$\Pi_\sigma^-\left(\mathrm{d}\Phi_Z^{-1}(\delta Z)\right) = P + \theta_2\left(\mathrm{ad}_P^2\right)\delta P - \mathrm{ad}_P\delta K \tag{23}$$

$$\Pi_\sigma^+\left(\mathrm{d}\Phi_Z^{-1}(\delta Z)\right) = \mathrm{dexp}_K^{-1}\left(\delta K + \mathrm{ad}_P\left(\theta_1\left(\mathrm{ad}_P^2\right)\delta P\right)\right) \tag{24}$$

We have here defined $\mathrm{d}\Phi^{-1} : \mathfrak{so}(n) \times \mathfrak{so}(n) \to \mathfrak{so}(n)$, but for our application we only need $\mathrm{d}\Phi^{-1} : \mathfrak{m}_Q \times \mathfrak{m}_{Q_1} \to \mathfrak{m}_Q$ for some $Q_1 \neq Q$. The advantage of this is that dexp_K^{-1} in (24) can be calculated using $k \times k$ matrices.

We now turn the attention to the problem of calculating analytic functions of ad_P^2. The derivation is a generalization of the special case $Q = I_{n,k}$ given in [12]. Let $P = \beta Q^T - Q\beta^T$ as defined above. We wish to calculate $\phi(\mathrm{ad}_P^2)\delta P$ for some $\delta P = \Pi_\sigma^-\delta Z$. Assume that β has full rank such that $-\beta^T\beta$ has k nonzero eigenvalues. Denote these eigenvalues by ν_i^2 and the corresponding eigenvectors by $\mathbf{x}_i$, i.e

$$-\beta^T\beta\mathbf{x}_i = \nu_i^2\mathbf{x}_i \quad i = 1, \cdots, k \tag{25}$$

We verify by multiplication that for $i = 1, \cdots, k$ we have

$$P\left(Q\mathbf{x}_i \pm \beta\mathbf{x}_i/\nu_i\right) = \pm\nu_i\left(Q\mathbf{x}_i \pm \beta\mathbf{x}_i/\nu_i\right) \tag{26}$$

thus $\pm\nu_i$ are eigenvalues of P and the corresponding (normalized) right eigenvectors are $\mathbf{v}_{\pm i} = \left(Qx_i \pm \beta x_i/\nu_i\right)/\sqrt{2}$. We have found $2k$ eigenvalues, and since P has maximal rank $2k$, these are all the nonzero eigenvalues. Similarly we find that the left eigenvalues and eigenvectors are $\pm\nu_i$ and $\mathbf{v}_{\pm i}^*$ (where $\mathbf{v}^*$ is the hermitian conjugate of $\mathbf{v}$). Now let Π_0 be the projection onto the zero-eigenspace, and let $\Pi_{\pm i}$ be the projections onto the nonzero eigenspaces, i.e

$$\begin{aligned} \Pi_{\pm i} &= \mathbf{v}_{\pm i}\mathbf{v}_{\pm i}^* \\ \Pi_0 &= I - \sum_{i=1}^{k}\left(\Pi_i + \Pi_{-i}\right) \end{aligned}$$

Observe now that

$$\begin{aligned}
\mathrm{ad}_P\left(\Pi_i \delta P \Pi_j\right) &= P\mathbf{v}_i\mathbf{v}_i^*\delta P\mathbf{v}_j\mathbf{v}_j^* - \mathbf{v}_i\mathbf{v}_i^*\delta P\mathbf{v}_j\mathbf{v}_j^* P \\
&= (v_i - v_j)\Pi_i\delta P\Pi_j
\end{aligned}$$

thus representing δP as $\delta P = \sum_{i,j}\Pi_i\delta P\Pi_j$, $i, j = 0, \pm 1, \cdots, \pm k$, we have that for each summand

$$\phi\left(\mathrm{ad}_P^2\right)\left(\Pi_{\pm i}\delta P\Pi_{\pm j}\right) = \phi\left(\pm(\nu_i - \nu_j)^2\right)\left(\Pi_i\delta P\Pi_j\right)$$

where ϕ is any analytic function. Since ad is linear in the second argument we now have a method for computing $\phi(\mathrm{ad}_P^2)$. At first glance this method may seem exhausting, but during calculation most parts cancel out, and the computation may be carried out with low cost. After a lengthy computation we arrive at the following: For $P = \beta Q^T - Q\beta^T$ and $\delta P = (\delta\beta)Q^T - Q(\delta\beta)^T$ and $\mathbf{x}_i, \nu_i$ defined in (25) the computation of $\phi(\mathrm{ad}_P^2)\delta P$ can be done with the formula

$$\phi\left(\mathrm{ad}_P^2\right)\delta P = MQ^T - QM^T \tag{27}$$

where M is the $n \times k$ matrix

$$M = \delta\beta(XCX^T) + \beta(XKX^T)$$

Here X is the $k \times k$ matrix with columns $\mathbf{x}_i$, C is the $k \times k$ diagonal matrix with entries

$$C_{ii} = \phi\left(\nu_i^2\right)$$

and finally, for A_{ij} being the ij-entry of $A = X^T\beta^T(\delta\beta)X$, K is the $k \times k$ matrix with entries

$$\begin{aligned}
K_{ij} &= \frac{A_{ij}}{\nu_i^2}\left(\phi\left(\nu_j^2\right) - \frac{1}{2}\phi\left((\nu_i - \nu_j)^2\right) - \frac{1}{2}\phi\left((\nu_i + \nu_j)^2\right)\right) \\
&\quad -\frac{1}{2}\frac{A_{ji}}{\nu_i\nu_j}\left(\phi\left((\nu_i - \nu_j)^2\right) - \phi\left((\nu_i + \nu_j)^2\right)\right)
\end{aligned}$$

In reaching our goal of nk^2 complexity algorithms, it is crucial that we newer form $n\times n$ matrices explicitely. In our applications (see algorithm 1 next section) we don't need all of $\mathrm{d}\Phi_Z^{-1}(\delta Z)$ we only need the part having an effect on our current point Q_n, that is the projection of $\mathrm{d}\Phi_Z^{-1}(\delta Z)$ onto the space $\mathfrak{m}_{Q_n}$. This observation allows us to work with $k \times k$ and $n \times k$ representations of the full $n \times n$ matrices involved, and this is the main tool in reaching the desired complexity.

4 Implementation and complexity

We take as the starting point the ordinary differential equation

$$\dot{Q} = H(Q,t)Q \quad Q(0) = Q_0 \tag{28}$$

where H is skew and $Q \in V_{n,k}$. We attempt to integrate this equation numerically using the theory previously presented, and will in this section provide algorithms and discuss their complexity. Let phi denote the function defined by

$$\mathrm{phi}(\alpha, \beta, Q) = \Phi(H(Q,t))Q \tag{29}$$

where α and β represents the projection of $H(Q,t)$ onto the space $\mathfrak{m}_Q$ according to (9). Further we let dphiinv be an order p approximation to the projection of $\mathrm{d}\Phi^{-1}_{H(Q_0,t_0)}H(Q_1,t_1)$ onto the space $\mathfrak{m}_{Q_0}$ where $Q_1 = \Phi(H(Q_0,t_0))Q$. Thus for computing dphiinv we need input α, β and Q_0 as in (29) and in addition the timestep t_1 and the order of approximation p. Given an s-stage order p Runge-Kutta method with Butcher tableau

$$
\begin{array}{c|cccc}
c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,s} \\
c_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,s} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s} \\
\hline
 & b_1 & b_2 & \cdots & b_s
\end{array}
$$

algorithm 1 represents one timestep with stepsize h using generalized polar coordinates.

Algorithm 1 One step using GPC with an s-stage Runge-Kutta method of order p, assuming the vector field is calculated inside the dphiinv procedure.

for $i = 1$ to s
  $\Theta_\alpha^{(i)} = h \sum_{j=1}^s a_{i,j} F_\alpha^{(j)}$
  $\Theta_\beta^{(i)} = h \sum_{j=1}^s a_{i,j} F_\beta^{(j)}$
  $[F_\alpha^{(i)}, F_\beta^{(i)}] = \text{dphiinv}(\Theta_\alpha^{(i)}, \Theta_\beta^{(i)}, Q_n, t_n + c_i h, p)$
end
$\Theta_\alpha = h \sum_{i=1}^s b_i F_\alpha^{(i)}$
$\Theta_\beta = h \sum_{i=1}^s b_i F_\beta^{(i)}$
$Q_{n+1} = \text{phi}(\Theta_\alpha, \Theta_\beta, Q_n)$

In the remaining part of this section we regard H as an operator rather than the matrix in (28), i.e. we let

$$H(u, v, t) := H(u, t)v \tag{30}$$

and we assume the operator H is given.

4.1 Algorithms

We present here `Matlab` functions for the coordinate map and the tangent map. The given algorithms are not optimal concerning floating point operation count, but they are to our knowledge optimal concerning simplicity. We discuss the possible cost improvements in subsection (4.2).

The calculation of the coordinate map is the simplest task using this scheme. We assume that α and β are given, and obtain the following algorithm, where ϕ_1 and ϕ_2 are defined in (13)

Algorithm 2 (Coordinate map) Calculates $Q_1 = \Phi(\beta Q_0^T - Q_0 \beta^T + Q_0 \alpha Q_0^T) Q_0$ given α, β and Q_0.

function $[Q_1]$=phi(α, β, Q_0)
% Set $\beta = \beta - Q_0 * (Q_0' * \beta)$ for greater stability
$[n, k] = \text{size}(\beta);$
$\Theta = -\beta^T \beta;$
$[X, \Lambda] = \text{eig}(\Theta);$
$\boldsymbol{\lambda} = \text{diag}(\Lambda);$
$v_1 = \text{zeros}(k);$ $v_2 = \text{zeros}(k);$
for $i = 1 : k$
  $v_1(:, i) = \phi_1(\boldsymbol{\lambda}(i)) * X(:, i);$
  $v_2(:, i) = \phi_2(\boldsymbol{\lambda}(i)) * X(:, i);$
end
$w = \text{expm}(\alpha);$
$Q_1 = \beta * (v_1 * X' * w) + Q * (v_2 * X' * w);$

By studying the formula (27) one may observe that we implicitly are computing the SVD of β using the *covariance matrix* $\beta^T\beta$. This procedure is known to be unstable for the small singular values. However, the terms containing the small singular values (i.e. the small eigenvalues of $\beta^T\beta$) give correspondingly small contributions to the overall result, so this does not seem to be a problem. The formula (27) will break down if the matrix β is not of full rank. If $\nu_i = 0$, we simply replace $\beta\mathbf{x}_i/\nu_i$ in (26) with a random vector which is orthogonalized with respect to all $\beta\mathbf{x}_j \quad j \neq i$, and then normalized.

In attempt to keep the presented algorithms simple we give here an algorithm based on the reduced SVD of β, i.e. $\beta = U\Sigma V^T$, which deals with rank deficiency without modifications. It is based on the observation that the eigenvalues ν_i in (26) are given $\nu_i = -\mathrm{i}\sigma_i$ (where σ_i are the singular values of β and $\mathrm{i} = \sqrt{-1}$) and that the eigenvectors $\mathbf{v}_{\pm i}$ corresponding to $\pm\nu_i$ are given $\mathbf{v}_{\pm i} = QV_i \pm \mathrm{i}U_i$ where V_i and U_i denotes the i-th column of the matrices U and V respectively.

Algorithm 3 (Analytic functions of ad^2**)** Calculates $\tilde{\beta}$ in the expression $\tilde{\beta}Q^T - Q\tilde{\beta}^T = f(\mathrm{ad}^2_{\beta Q^T - Q\beta^T}((\delta\beta)Q^T - Q(\delta\beta)^T))$ where f is any given analytic function. This algorithm uses the singular value decomposition of β rather than the eigenvalue decomposition of $-\beta^T\beta$

function $[\tilde{\beta}] = \mathrm{fad2}(\beta, \delta\beta, f)$;
$[n, k] = \mathrm{size}(\beta)$;
$[U, \Sigma, V] = \mathrm{svd}(\beta, 0)$;
$\boldsymbol{\sigma} = \mathrm{diag}(\Sigma)$;
$A = (U^T * \delta\beta) * V$;
$K = \mathrm{zeros}(k); \quad C = \mathrm{zeros}(k)$;
for $i = 1 : k$
  $C(i,i) = \mathrm{feval}(f, -\boldsymbol{\sigma}(i)^2)$;
  for $j = 1 : k$
    $K(i,j) = \frac{1}{2}\mathrm{feval}\left(f, -(\boldsymbol{\sigma}(i) - \boldsymbol{\sigma}(j))^2\right) * (A(i,j) + A(j,i))...$
      $+\frac{1}{2}\mathrm{feval}\left(f, -(\boldsymbol{\sigma}(i) + \boldsymbol{\sigma}(j))^2\right) * (A(i,j) - A(j,i)) - \mathrm{feval}\left(f, -\boldsymbol{\sigma}(j)^2\right) * A(i,j)$;
  end
end
$\tilde{\beta} = \delta\beta * (V * C * V^T) + \beta * (V * K * V^T)$

Given a function for computing analytic functions of ad^2, the algorithm for the tangent map is straight forward just keeping in mind that we only work with the "α" and "β" representations of the $n \times n$ matrices involved. We assume that the vectorfield function H, the dexpinv function and the functions θ_1 and θ_2 (21) are given.

Algorithm 4 (Tangent map) Computes $\tilde{\alpha}$ and $\tilde{\beta}$ in $\tilde{H} = \tilde{\beta}Q^T - Q\tilde{\beta}^T + Q\tilde{\alpha}Q^T$ where $\tilde{H}$ is a p order approximation to the projection of $\mathrm{d}\Phi^{-1}_{H(Q_0,t_0)}(H(Q_1,t))$ onto the space $\mathfrak{m}_{Q_0}$ for $Q_1 = \Phi(H(Q_0,t_0))Q_0$.

function $[\tilde{\alpha}, \tilde{\beta}] = \mathrm{dphiinv}(\alpha, \beta, Q_0, t, p)$
$Q_1 = \mathrm{phi}(\alpha, \beta, Q_0)$;
$u = H(Q_1, Q_0, t)$;
$\delta\alpha = Q_0^T * u$;
$\delta\beta = u - Q_0 * \delta\alpha_0$;
$v = H(Q_1, \beta, t)$;
$w = \beta_0 * \delta\alpha - v + Q_0 * (Q_0^T * v)$;
$m = \beta_0^T * \mathrm{fad2}(\beta, \delta\beta, \theta_1)$;
$\tilde{\alpha} = \mathrm{dexpinv}(\alpha_0, \delta\alpha_0 - m + m^T, p)$;
$\tilde{\beta} = \delta\beta_0 + \mathrm{fad2}(\beta, \delta\beta, \theta_2) - w$;

4.2 Complexity

We assume that $n \gg k$ and only comment the asymptotic complexity, thus we only count operations of the leading term nk^2. In the coordinate map there are three multiplications involving $n \times k$ matrices and thus the algorithm requires $\sim 6nk^2$ flops. ($\sim 10nk^2$ for greater stability). For the calculation of the tangent map we obtain considerable savings if the fad2 algorithm is incorporated into the dphiinv algorithm. If we in addition use the eigenvalue decomposition of $-\beta^T\beta$ (rather than the SVD as in algorithm 3) we can assume that this has all ready been calculated in the phi algorithm. In this way we count a total of seven multiplications involving $n \times k$ matrices and the total cost is $\sim 14nk^2$. In addition we need two evaluations of $H(u, v, t)$.

5 Numerical experiments

5.1 Comparing GPC with the exponential map

In this example we consider an ODE

$$\dot{Y}(t) = AY(t) \quad Y(0) = Y_0$$

where $Y \in \mathbb{R}^{100\times 4}$ and $A \in \mathbb{R}^{100\times 100}$ is a constant random 5-diagonal matrix. We choose randomly Q_0 and R_0 and set $Y_0 = Q_0R_0$. We attempt to calculate the Q-factor in the QR-factorization of $Y(1)$ which then is compared to the "exact" factor obtained by the QR-factorization of $\exp(A)Y_0$. The restriction of the system to the Stiefel manifold is given

$$\dot{Q} = H(Q)Q, \qquad Q(0) = Q_0$$

where

$$H(Q) = \beta Q^T - Q\beta^T + Q\alpha Q^T$$

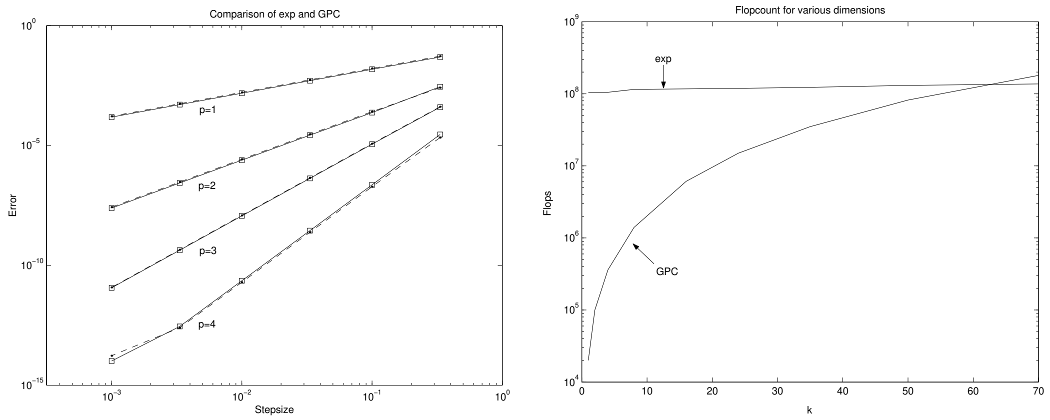
with $\beta = (I - QQ^T)AQ$ and $\alpha = \mathrm{tril}(Q^TAQ) - \mathrm{tril}(Q^TAQ)^T$. The system is integrated from $t = 0$ to $t = 1$ using methods of order 1,2,3 and 4, and with various stepsizes. Both the maps exp and GPC maintain orthogonality to machine accuracy, and surprisingly, as seen in figure 1(a), the difference in performance of the two maps is negligible. In figure 1(b) we give flop counts for a fourth order method, integrating the system varying k in the $n \times k$ matrix Y but keeping $n = 100$ fixed. As expected the usage of GPC shows huge savings compared to exp for small values of k, but as k increases, the large number of k^3 operations closes the gap. Although the computation of $H(u)v$ in this example is done in $O(nk^2)$ flops, it is the major contributor to the cost of GPC for small values of k.

5.2 Computing Lyapunov exponents

In this example we consider a test problem for computing Lyapunov exponents. The problem has been considered in [4], [1] and [2] and references therein. The system

$$\begin{aligned}
\ddot{y} &= -\alpha\left(y^2 - 1\right)\dot{y} - \omega y \\
\ddot{x}_1 &= -d_1\dot{x}_1 - \beta\left[V'\left(x_1 - x_n\right) - V'\left(x_2 - x_1\right)\right] + \sigma y \\
\ddot{x}_i &= -d_i\dot{x}_i - \beta\left[V'\left(x_i - x_{i-1}\right) - V'\left(x_{i+1} - x_i\right)\right]\ ,\ i = 2, \cdots, n
\end{aligned}$$

describes a ring of n damped oscillators with amplitude x_i with $x_{n+1} = x_1$. The ring is forced externally by $y(t)$. The parameters $\alpha, \beta, \omega, \sigma$ and the potential function $V(x)$ are chosen as in [1], i.e $V(x) = x^2/2 + x^4/4$, $\alpha = 1$, $\beta = 1$, $\omega = 1$ and $\sigma = 4$. The damping parameters are set to $d_i = 0.0125$ for i odd, and $d_i = 0.0075$ for i even. The experiment is done with $n = 5$. A randomly chosen $\mathbf{x}_0$ is used as initial value, and the system is integrated from $t = 0$ to $t = 4000$ with stepsize 0.01 using the classical fourth order Runge Kutta method. Let $A(t)$ be the linearization of the



(a) (b)

Figure 1: (a) The usage of GPC (dashed line) yields almost exactly the same result as using an RKMK method with the exponential map (solid line). (b) Usage of GPC shows great savings compared to exp for $n \gg k$.

system along the computed trajectory $\mathbf{x}(t)$, and consider the following differential equation on the Stiefel manifold

$$\dot{Q} = H(Q)Q \qquad Q(0) = I_{12,4}$$

where H is defined as in the previous example. The 4 largest Lyapunov exponents of the system are now given (see [4] for details)

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \int_0^t B_{ii}(s)ds \quad k = 1, 2, 3, 4$$

where $B = Q^T AQ - \alpha$. The integral is approximated using the trapezoidal rule. The computed four largest Lyapunov exponents at various time stages are plotted in figure 2. Again the maps exp and GPC yield approximately the same answer as shown in table 1.

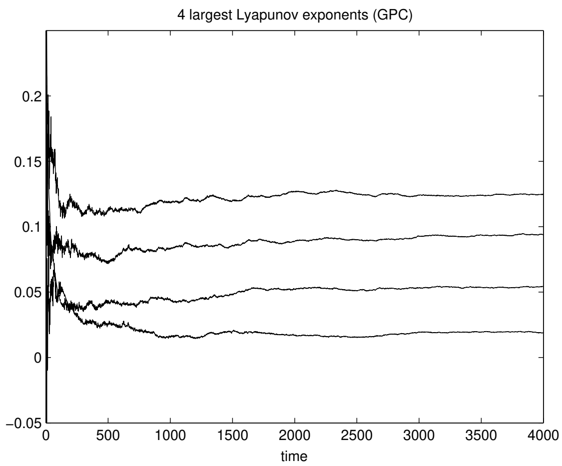


Figure 2: The four largest Lyapunov exponents using GPC. The usage of RKMK with the exponential map gave the same plot.

Table 1: Computed Lyapunov exponents using GPC and exp

	Value GPC	Value exp	max last 500	min last 500
λ_1	0.12471298	0.12470863	$+0.0006$	-0.0012
λ_2	0.09391670	0.09391934	$+0.0010$	-0.0011
λ_3	0.05417468	0.05417294	$+0.0003$	-0.0013
λ_4	0.01868826	0.01868608	$+0.0016$	-0.0002

6 Concluding remarks

In this paper we have presented a $O(nk^2)$ complexity Lie group method for numerical integration on the Stiefel manifold. The order of complexity is obtained by choosing a base point dependent type of generalized polar coordinates which is a first order approximation to the matrix exponential. By the paper of Muthe-Kaas and Zanna [12] we are able to compute any given order approximation to the derivative of the coordinate map, and thus any order p Runge-Kutta method can be used as a base to obtain an order p Lie group method.

Lie group methods for numerical integration on the Stiefel manifold can without precautions force an arithmetic complexity of order n^3, and this has been the main drawback. In this paper we have shown that a reduction in complexity is possible, and that the resulting method, at least for the examples presented, have the same qualitative behavior as the original RKMK method.

The representation of the Stiefel manifold as a quotient space of the orthogonal group was of crucial importance in the derivation of a convenient representation of the tangentspace. As aslo commented in [1] there is no doubt that knowledge of the differential geometry of the Stiefel manifold is of great help when constructing numerical methods, and it is an interesting question whether this carries over to other homogenous manifolds. With the aid of this theory we were able to represent any skew symmetric matrix by a canonical representative from the horizontal bundle at a given point on the manifold. This representative, typically being of low rank, can again be represented using only $n \times k$ and $k \times k$ matrices allowing all our computations to be done with $O(nk^2)$ complexity. We defined a projection onto the horizontal bundle which for skew symmetric matrices is canonical, but generally leaving us with choices corresponding to different ways of obtaining orthonormal factors of an arbitrary matrix. One such instance giving a (Lie type) differential equation for the continuous QR decomposition. We are not aware if there exists a similar equation for the polar decomposition, although the existence of a continuous polar decomposition (among other decompositions) is certified in [3].

Acknowledgments

I would like to thank Hans Z. Munthe-Kaas for suggesting an investigation on this topic and for his valuable help and comments.

References

[1] T. J. Bridges and S. Reich. Computing Lyapunov exponents on a Stiefel manifold. Technical report, University of Surrey, April 2000.

[2] E. Celledoni and B. Owren. A class of low complexity intrinsic schemes for orthogonal integration. University of Trondheim, February 2001.

[3] L.Dieci and T.Eirola. On smooth decompositions of matrices. *SIAM J.Matrix Anal. Appl.* 20(3):800-819, 1999.

[4] L. Dieci, E. S. Van Vleck. Computation of a few Lyapunov exponents for continous and discrete dynamical systems. *Appl. Num. Math.*, 17:275-291, 1995.

[5] L.Dieci, R. D. Russel and E. S. Van Vleck. On the computation of Lyapunov exponents for continuous dynamical systems. *SIAM J. Numer. Anal.,* 34:402-423, 1997.

[6] A. Edelman, T.A. Arias and S. T. Smith. The geometry of algorithms with ortogonality constrains. *SIAM J. Matrix Anal. Applic.,* 20(2):303-353, 1999.

[7] K.Engø. On the construction of geometric integrators in the RKMK class. *BIT* 40(1):041-061, 2000.

[8] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett and A. Zanna. Lie-group methods. *Acta Numerica 9,* CUP, 215-365, 2000.

[9] S.Kobayashi and K.Nomizu. *Foundations of Differential Geometry, Volume I* (John Wiley & sons, New York, 1963)

[10] H.Z. Munthe-Kaas. High order Runge-Kutta methods on manifolds. *Appl. Num. Math.* 29:115-127, 1999.

[11] H.Z. Munthe-Kaas, G.R.W. Quispel and A. Zanna. The polar decomposition on Lie groups with involutive automorphisms. Technical report no. 191. University of Bergen, May 2000.

[12] H.Z.Munthe-Kaas, A. Zanna. Lie group methods based on generalized polar coordinates. University of Bergen, In preparation.

[13] S.T. Smith. *Geometric Optimization Methods for Adaptive Filtering.* Phd thesis. Harvard University, May 1993.

[14] A.Zanna and H.Z.Munthe-Kaas. Generalized polar decompositions for the approximation of the matrix exponential. Report no. 200. University of Bergen, August 2000.