REPORTS
IN
INFORMATICS

ISSN 0333-3590

Generalized Polar Coordinates on Lie groups and Numerical Integrators

S. Krogstad, H. Z. Munthe-Kaas, A. Zanna

REPORT NO 244 **April 2003**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/pdf/2003-244.pdf

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

Generalized Polar Coordinates on Lie groups and Numerical Integrators

S. Krogstad, H. Z. Munthe-Kaas, A. Zanna

Abstract

Motivated by recent developments in numerical Lie group integrators, we introduce a family of local coordinates on Lie groups denoted *generalized polar coordinates*. Fast algorithms are derived for the computation of the coordinate maps, their tangent maps and the inverse tangent maps. In particular we discuss algorithms for all the classical matrix Lie groups and optimal complexity integrators for n-spheres.

1 Introduction

Lie group methods for integration of differential equations has been an active area of research over the last decade [7, 20, 24, 17, 22]. Consider the family of integrators based on local coordinates as presented in [12] (see Algorithm 1). These methods are expressed in terms of a local coordinate map Φ from a Lie algebra to a Lie group and the inverse tangent map of Φ.

Analytic coordinate maps include the exponential map [22], the Cayley map, and more generally diagonal Padé approximants to the exponential. It is well known that for certain groups (e.g. $\mathrm{SL}(n)$), the only analytic map from the algebra to the group is the exponential mapping [18]. Matrix splitting techniques yield non-analytic coordinate maps. Among these *coordinates of the second kind*, studied by Owren and Marthinsen [25], show excellent computational cost for certain groups. Unfortunately, it is not known how this approach can be applied to e.g. the real orthogonal group.

In this paper an other family of coordinates based on matrix splittings is studied. By recursively applying *generalized polar decompositions* of the Lie algebra [27, 23], we obtain coordinates on all the classical matrix groups, where both the coordinate maps and the (forward and inverse) tangent maps can be computed efficiently.

Let us briefly review the generalized polar coordinates as defined in Section 3. Consider a nested sequence of Lie algebras

$$\mathfrak{g} = \mathfrak{g}_0 \supset \mathfrak{g}_1 \supset \cdots \supset \mathfrak{g}_k$$

derived from a sequence of linear maps $\sigma_i : \mathfrak{g}_i \to \mathfrak{g}_i$, $i = 0, \ldots, k-1$, such that the following relations hold

$$\begin{aligned} \sigma_i^2 &= I \\ \sigma_i([X,Y]) &= [\sigma_i(X), \sigma_i(Y)] \\ \mathrm{Range}\left(\frac{1}{2}(I+\sigma_i)\right) &= \mathfrak{g}_{i+1}. \end{aligned}$$

An arbitrary element $Z \in \mathfrak{g}$ is decomposed as

$$Z = P_0 + P_1 + \cdots + P_{k-1} + K_k, \tag{1.1}$$

where the components are computed as follows. Let $K_0 = Z$ and for $i = 0, \dots, k-1$ let

$$\begin{aligned} K_{i+1} &= \frac{1}{2}(I+\sigma_i)K_i \in \mathfrak{g}_{i+1} \\ P_i &= K_i - K_{i+1} = \frac{1}{2}(I-\sigma_i)K_i. \end{aligned}$$

The coordinate maps to be studied in this paper are of the form

$$\Phi(Z) = \exp(P_0)\cdot\exp(P_1)\cdots\exp(P_{k-1})\cdot\exp(K_k). \tag{1.2}$$

It is assumed that the computations in the last Lie algebra $\mathfrak{g}_k$ can be done fast, either because of low dimensionality or because it has a special structure (diagonal or block diagonal matrices). The computations involving the matrices P_i and the tangent maps are relying on the structure of 2-cyclic matrices to be discussed in the sequel.

The paper is structured as follows. In Section 2 we review basic results of linear algebra, matrix Lie theory and numerical Lie group integrators. Section 3 introduces the generalized polar coordinates, and develops the theory of their tangent maps. In Section 4 the classical matrix groups, symmetric spaces and spheres are discussed in detail, and in Section 5 we present numerical examples.

2 Preliminaries

2.1 Coordinates in Lie group integrators

In order to motivate the theory in the sequel, we will briefly review a class of numerical integrators introduced in [21, 22]. We will present the theory in the concrete context of matrix Lie groups. The generalization to general Lie groups is discussed in [16].

Let $\mathfrak{g}$ denote a *matrix Lie algebra*, i.e. a family of square matrices closed under linear combinations and matrix commutators,

$$[A,B] = AB - BA.$$

Let G denote the *Lie group* of $\mathfrak{g}$, defined as the set of matrices obtained by taking matrix exponentials of $\mathfrak{g}$ and products of these exponentials. The Lie group is closed under matrix products and matrix inversions. Let $\mathcal{M}$ denote a linear space and $\cdot : G\times\mathcal{M} \to \mathcal{M}$ an *action* of G on $\mathcal{M}$, defined as a map satisfying

$$g\cdot(h\cdot y) = (gh)\cdot y, \quad \text{for all } g,h \in G,\, y\in\mathcal{M}.$$

The action induces a product $\cdot : \mathfrak{g}\times\mathcal{M} \to T\mathcal{M}$, where $T\mathcal{M}$ denotes the tangents of $\mathcal{M}$, via

$$V\cdot y = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} \exp(tV)\cdot y.$$

We consider differential equations evolving on $\mathcal{M}$, written in the form [22]

$$y'(t) = f(y(t))\cdot y, \tag{2.1}$$

where $y(t) \in \mathcal{M}$ and $f : \mathcal{M} \to \mathfrak{g}$.

Let $\Phi : \mathfrak{g} \to G$ denote a smooth mapping from a Lie algebra into a Lie group such that $\Phi(0) = e$, where e is the identity in G. Let $d\Phi_Z$ denote the right trivialized tangent of Φ at a point $Z \in \mathfrak{g}$, i.e.

$$\left.\frac{\partial}{\partial s}\right|_{s=0} \Phi(Z + s\delta Z) = d\Phi_Z(\delta Z)\Phi(Z).$$

It follows easily that $d\Phi_Z$ is a linear map from $\mathfrak{g}$ to itself. We assume that $d\Phi_0 = I$, thus $d\Phi_Z$ is invertible for Z sufficiently close to 0. The map Φ defines a diffeomorphism between a neighbourhood of $0 \in \mathfrak{g}$ and a neighbourhood of $e \in G$. Via the translations on G, we may extend Φ to an atlas of coordinate charts covering the whole of G. Using the action of of G on $\mathcal{M}$, we may also via Φ obtain a coordinate atlas on $\mathcal{M}$. For numerical algorithms based on such coordinates, it is often essential that we can compute both Φ and the inverse tangent map $d\Phi_Z^{-1}$ efficiently.

As an example, consider the class of numerical Lie group integrators for (2.1), introduced and developed in [22, 12]. Given $y_n \in \mathcal{M}$ and a timestep $h \in \mathbb{R}$. Given $\{a_{i,j}\}_{i,j=1}^s$ and $\{b_j\}_{j=1}^s$, the coefficients of an s-stage Runge-Kutta method. We step from $y_n \approx y(t_n)$ to $y_{n+1} \approx y(t_n + h)$ as:

Algorithm 1

for $i = 1, s$,
  $U_i = \sum_{j=1}^s a_{i,j}\tilde{K}_j$
  $K_i = h \cdot f(\Phi(U_i)\cdot y_n)$
  $\tilde{K}_i = d\Phi_{U_i}^{-1}(K_i)$
end
$y_{n+1} = \Phi(\sum_{j=1}^s b_j\tilde{K}_j)\cdot y_n$

Various coordinate maps Φ have been proposed and studied in the literature. The exponential mapping is used in [22] and the Cayley map in [20].

Owren and Marthinsen [25] develop the theory of *coordinates of the second kind*. Given a basis $\{V_i\}$ for a d-dimensional Lie algebra $\mathfrak{g}$, an element $Z = \sum_{i=1}^d z^i V_i$ maps to

$$\Phi(Z) = \exp(z^1V_1)\cdot\exp(z^2V_2)\cdots\exp(z^dV_d).$$

Owren and Marthinsen introduce special classes of 'nice' bases, so-called Admissible Ordered Bases (AOBs), and show that for such bases the map $\Phi(Z)$ and $d\Phi_Z{}^{-1}$ can be computed in $\mathcal{O}(d^{3/2})$ operations, $d = \dim\mathfrak{g}$. Using the representation theory of semisimple Lie algebras, they show that for certain semisimple Lie algebras, AOBs can be obtained from Chevalley bases. In the cases where AOBs are found, they report favourable numerical experiments indicating that the resulting numerical algorithms are between two and six times faster than corresponding algorithms based on using $\Phi(Z) = \exp(Z)$. Unfortunately, it is not known if AOBs can be found for all classical matrix Lie groups. In particular there are still open problems with several of the real matrix groups, such as the real orthogonal group $SO(n,\mathbb{R})$ with algebra $\mathfrak{so}(n,\mathbb{R})$, consisting of real skew symmetric matrices.

In this note we will introduce coordinates based on generalized polar decompositions of $\mathfrak{g}$, and we will show that this leads to fast computable coordinates for many Lie algebras, among these all the classical matrix Lie algebras. Furthermore, the theory of generalized polar coordinates is considerably simpler than the theory of second kind coordinates.

The basis for our approach is some results from the theory of symmetric spaces, as given in [23]. We will now review some linear algebra and basic elements of the theory of symmetric spaces needed for the present purpose.

2.2 Projections, involutions and 2-cyclic matrices

By a *projection matrix* on a vector space $\mathcal{V}$ we mean a linear map $\Pi : \mathcal{V} \to \mathcal{V}$ such that $\Pi^2 = \Pi$. Unless explicitly stated we will not require projections to be orthogonal (i.e. $\Pi^T = \Pi$). By an *involutive matrix* we mean a linear map $S : \mathcal{V} \to \mathcal{V}$ such that $S^2 = I$. These two concepts are naturally linked by the following lemma, whose proof is trivial.

Lemma 2.1 *To any projection matrix* Π *there corresponds an involutive matrix* $S = I - 2\Pi$. *Conversely, to any involutive* S *there corresponds two projection matrices* Π_S^- *and* Π_S^+ *defined by*

$$\Pi_S^- = \frac{1}{2}(I - S) \tag{2.2}$$

$$\Pi_S^+ = \frac{1}{2}(I + S). \tag{2.3}$$

These satisfy the following relations

$$\Pi_S^- + \Pi_S^+ = I \tag{2.4}$$

$$\Pi_S^- \Pi_S^+ = \Pi_S^+ \Pi_S^- = 0 \tag{2.5}$$

$$S\Pi_S^- = -\Pi_S^- \tag{2.6}$$

$$S\Pi_S^+ = \Pi_S^+. \tag{2.7}$$

Thus $\mathcal{V}$ splits in the direct sum of two subspaces, the ± 1 eigenspaces of S, where $\Pi_S^\pm$ are projections onto these. Note that if S is involutive then also $-S$ is involutive, the latter corresponding to the opposite identification of the $+1$ and -1 eigenspaces. Thus, at the moment there seems to be no fundamental difference between these two subspaces. We will, however, later return to involutive automorphisms on Lie algebras where the $+1$ and -1 eigenspaces play fundamentally different roles.

Definition 2.1 A matrix $K : \mathcal{V} \to \mathcal{V}$ is *block diagonal* with respect to an involution S on $\mathcal{V}$ if

$$SKS = K, \tag{2.8}$$

and a matrix $P : \mathcal{V} \to \mathcal{V}$ is *2-cyclic* with respect to S if

$$SPS = -P. \tag{2.9}$$

Any matrix M can be split in a sum of a 2-cyclic P and a block diagonal K,

$$M = P + K, \tag{2.10}$$

where

$$P = \frac{1}{2}(M - SMS) \tag{2.11}$$

$$K = \frac{1}{2}(M + SMS). \tag{2.12}$$

To understand the structure of these matrices, it is convenient to represent linear operators $M : \mathcal{V} \to \mathcal{V}$ in 2×2 block partitioned form in the following manner. The matrix M splits naturally in 4 parts:

$$M = \left(\Pi_S^- + \Pi_S^+\right) M \left(\Pi_S^- + \Pi_S^+\right) = \Pi_S^- M \Pi_S^- + \Pi_S^- M \Pi_S^+ + \Pi_S^+ M \Pi_S^- + \Pi_S^+ M \Pi_S^+. \tag{2.13}$$

The *partitioning of* M *with respect to* S is defined by dividing $\mathcal{V}$ in an upper block corresponding to the range of Π_S^- and a lower block corresponding to the range of Π_S^+. Thus

$$M = \begin{pmatrix} M^{--} & M^{-+} \\ M^{+-} & M^{++} \end{pmatrix}, \tag{2.14}$$

where $M^{ij} = \Pi_S^i M \Pi_S^j$ restricted to the appropriate subspaces. In partitioned form S becomes

$$S = \begin{pmatrix} -I & 0 \\ 0 & I \end{pmatrix}.$$

Thus, K consists of the diagonal blocks of M, while P is the off diagonal blocks.

Efficient computation of analytic functions of 2-cyclic matrices will be crucial to our algorithms.

Theorem 2.2 *Let $SPS = -P$, where $S^2 = I$. Let $\Theta = P^2\Pi_S^-$. For an analytic function $\psi(x)$ we have*

$$\psi(P) = \psi(0)I + \psi_1(\Theta)P + P\psi_1(\Theta) + P\psi_2(\Theta)P + \psi_2(\Theta)\Theta, \tag{2.15}$$

where

$$\psi_1(x) = \frac{1}{2\sqrt{x}}\left(\psi(\sqrt{x}) - \psi(-\sqrt{x})\right) \tag{2.16}$$

$$\psi_2(x) = \frac{1}{2x}\left(\psi(\sqrt{x}) + \psi(-\sqrt{x}) - 2\psi(0)\right) \tag{2.17}$$

and where we define $\Theta^0 \equiv \Pi_S^-$.

Proof. Involution of S implies $SP = -PS$ hence $\Pi_S^{\pm}P = P\Pi_S^{\mp}$. By induction it is now easy to verify that for any $k \geq 0$ we have

$$P^{2k+1} = \Theta^k P + P\Theta^k, \quad P^{2k+2} = \Theta^{k+1} + P\Theta^k P.$$

Letting $\psi(x) = \sum_{i=0}^{\infty} \alpha_i x^i$, we find from these formulae that

$$\psi(P) = \alpha_0 I + \sum_{k=0}^{\infty} \alpha_{2k+1}\Theta^k P + P\sum_{k=0}^{\infty} \alpha_{2k+1}\Theta^k + \sum_{k=0}^{\infty} \alpha_{2k+2}\Theta^{k+1} + P\sum_{k=0}^{\infty} \alpha_{2k+2}\Theta^k P.$$

We define $\psi_1(x) = \sum_{k=0}^{\infty} \alpha_{2k+1}x^k$ and $\psi_2(x) = \sum_{k=0}^{\infty} \alpha_{2k+2}x^k$ and derive (2.16) and (2.17) by straightforward manipulation of the series for $\psi(x)$. □

Note that if P is partitioned with respect to S as

$$P = \begin{pmatrix} 0 & B \\ A & 0 \end{pmatrix}, \tag{2.18}$$

then

$$\Theta = P^2\Pi_S^- = \begin{pmatrix} BA & 0 \\ 0 & 0 \end{pmatrix}. \tag{2.19}$$

Thus

$$\psi_i(\Theta) = \begin{pmatrix} \psi_i(BA) & 0 \\ 0 & \psi_i(0)I \end{pmatrix}.$$

Since BA is a $p \times p$ matrix, $p = \text{Rank}(\Pi_S^-)$, we are mainly interested in cases where p is small, and the case $p = 1$ is particularly important.

Corollary 2.2.1 *If $p = 1$ then $\Theta = \theta\Pi_S^-$, for a scalar θ, and*

$$\psi(P) = \psi(0)I + \psi_1(\theta)P + \psi_2(\theta)P^2, \tag{2.20}$$

where ψ_1 and ψ_2 are given in (2.16)-(2.17).

Proof. If $\Theta = \theta \Pi_S^-$ then

$$\psi_1(\Theta)P + P\psi_1(\Theta) = \psi_1(\theta)\left(\Pi_S^- P + P\Pi_S^-\right) = \psi_1(\theta)P.$$

Similarly

$$P\psi_2(\Theta)P + \psi_2(\Theta)\Theta = \psi_2(\theta)\left(P\Pi_S^- P + \Pi_S^- P^2\right) = \psi_2(\theta)P^2.$$

□

2.3 Computation of analytic matrix functions

Let $A : \mathcal{V} \to \mathcal{V}$ be a matrix with a (small) number of different eigenvalues $\lambda_1, \ldots, \lambda_d$. Let $\psi(x)$ be an analytic function. For later application, we will discuss some algorithms for computing

$$\boldsymbol{w} = \psi(A)\boldsymbol{v} \tag{2.21}$$

for an arbitrary vector $\boldsymbol{v} \in \mathcal{V}$.

2.3.1 Via eigenspace projections

In the following we assume that A is non-defect (all Jordan blocks of size 1). For $i = 1, \ldots, d$, let Π_i denote projection onto the eigenspace corresponding to λ_i. Then

$$\begin{aligned} \sum_{i=1}^{d} \Pi_i &= I \\ A\Pi_i &= \Pi_i A = \lambda_i \Pi_i. \end{aligned}$$

Thus (2.21) can be computed as

$$\boldsymbol{w} = \psi(A)\boldsymbol{v} = \sum_{i=1}^{d} \psi(\lambda_i)\Pi_i \boldsymbol{v}. \tag{2.22}$$

The eigenspace projections can be expressed in terms of the left and right eigenvectors. Let $\boldsymbol{x}_{i,k}$ and $\boldsymbol{y}_{i,k}$ denote all the right and left eigenvectors of A, where $i = 1, \ldots, d$ and $k = 1, \ldots, \mathrm{Rank}\,(\Pi_i)$. These can be chosen such that

$$A\boldsymbol{x}_{i,k} = \lambda_i \boldsymbol{x}_{i,k} \tag{2.23}$$

$$\boldsymbol{y}_{i,k}^T A = \lambda_i \boldsymbol{y}_{i,k}^T \tag{2.24}$$

$$\boldsymbol{y}_{i,k}^T \boldsymbol{x}_{j,l} = \delta_{i,j}\delta_{k,l}. \tag{2.25}$$

In other words, if $\boldsymbol{x}_{i,k}$ are the columns of the eigenvector matrix X, then $\boldsymbol{y}_{j,l}^T$ are the rows of X^{-1}. We have

$$\Pi_i = \sum_k \boldsymbol{x}_{i,k}\boldsymbol{y}_{i,k}^T. \tag{2.26}$$

We will return to other ways of representing these projections in the sequel.

2.3.2 Via the minimal polynomial

Assuming A is non-defect, the d-degree polynomial

$$q(x) = \Pi_{j=1}^{d}(x-\lambda_j)$$

is the minimal polynomial of A, i.e. the lowest degree monic polynomial for which $q(A) = 0$. Let $r(x) = \psi(x) \bmod q(x)$ be the division remainder, defined as the degree $d-1$ polynomial such that

$$\psi(x) = q(x)s(x) + r(x),$$

where $s(x)$ does not have singularities in λ_j. If $\psi(x)$ is a polynomial, the remainder $r(x)$ can be computed by polynomial division. In the general analytic case, $r(x)$ can be found from polynomial interpolation in the d points

$$r(\lambda_i) = \psi(\lambda_i), \quad \text{for } i = 1, \dots, d.$$

Since $q(A) = 0$ we see that $\boldsymbol{w} = \psi(A)\boldsymbol{v} = r(A)\boldsymbol{v}$ can be computed by the work of $d-1$ matrix-vector products $A\boldsymbol{v}$.

In the general case, where A might be defect, let m_j denote the size of the largest Jordan block associated with the eigenvalue λ_j. The minimal polynomial is now given as

$$q(x) = \Pi_{j=1}^{d}(x-\lambda_j)^{m_j}.$$

We obtain $r(x)$ by Hermite interpolation. In λ_j we let $r(x)$ interpolate $\psi(x)$ and its derivatives up to order $m_j - 1$. The Newton form of $r(x)$ is convenient to work with.

Let divided differences be defined as

$$\psi[\lambda_i] = \psi(\lambda_i) \tag{2.27}$$

$$\psi[\lambda_i, \dots, \lambda_{i+k}] = \frac{\psi[\lambda_{i+1}, \dots, \lambda_{i+k}] - \psi[\lambda_i, \dots, \lambda_{i+k-1}]}{\lambda_{i+k} - \lambda_i} \quad \text{for distinct } \lambda_j \tag{2.28}$$

$$\psi[\lambda_j, \lambda_j, \dots, \lambda_j] = \frac{1}{(m_j-1)!}\psi^{(m_j-1)}(\lambda_j) \quad \text{for } \lambda_j \text{ repeated } m_j \text{ times.} \tag{2.29}$$

Then $r(x)$ is given as

$$r(x) = \psi[x_1] + (x-x_1)\psi[x_1,x_2] + \cdots + (x-x_1)(x-x_2)\cdots(x-x_{m-1})\psi[x_1,x_2,\dots,x_m], \tag{2.30}$$

where $m = \sum_{i=1}^{d} m_i$ and

$$\{x_1, x_2, \dots, x_m\} = \{\lambda_1, \dots, \lambda_1, \lambda_2, \dots, \lambda_2, \dots, \lambda_d, \dots, \lambda_d\} \quad (\lambda_i \text{ repeated } m_i \text{ times}).$$

Using this form, we obtain the following simple algorithm to compute $z = \psi(A)\boldsymbol{v} = r(A)\boldsymbol{v}$ using $m-1$ matrix vector products.

Algorithm 2 (Computing $\boldsymbol{w} = \psi(A)\boldsymbol{v}$)

```
w := ψ[x_1] v
for  i = 1, ..., m − 1
     v := A v − x_i v
     w := w + ψ[x_1, ..., x_{i+1}] v
end
```

2.3.3 Via Schur decompositions

If A is far away from a normal matrix, or if it is defect, the eigenspace projection approach is numerically unstable. A better class of algorithms is based on transforming A to an upper triangular matrix by the Schur decomposition. Functions of upper triangular matrices can be computed by the algorithm of Parlett [13], and a general purpose algorithm including these steps is given in [8]. We return to this approach in Section 3.3.

2.4 Involutive automorphisms on Lie algebras

An *involutive automorphism* on a Lie algebra $\mathfrak{g}$ is an involutive map $\sigma : \mathfrak{g} \to \mathfrak{g}$ such that

$$\sigma([U,V]) = [\sigma(U), \sigma(V)]. \tag{2.31}$$

Corresponding to the automorphism σ on $\mathfrak{g}$ there is an automorphism on the Lie group G, which plays an important role in the theory of symmetric spaces. In this paper we will, however, not need this automorphism on G and we omit this from the discussion.

Note that if σ is an automorphism, then $-\sigma$ is *not* an automorphism. Thus in this case we can clearly distinguish between the $+1$ and -1 eigenspaces of σ, they play a fundamentally different role in the theory. Let $\Pi_\sigma^\pm$ be projections on $\pm$ eigenspaces of σ, given in (2.2) - (2.3). Denote these spaces by

$$\begin{aligned}\mathfrak{p} &= \mathrm{Range}(\tfrac{1}{2}(I-\sigma)) = \{\, P \in \mathfrak{g} \mid \sigma(P) = -P \,\}\\ \mathfrak{k} &= \mathrm{Range}(\tfrac{1}{2}(I+\sigma)) = \{\, K \in \mathfrak{g} \mid \sigma(K) = K \,\}.\end{aligned}$$

The subspace $\mathfrak{k}$ is a Lie subalgebra of $\mathfrak{g}$, while $\mathfrak{p}$ forms a *Lie triple system*, meaning that it is closed under *double brackets*

$$[P_1, [P_2, P_3]] \in \mathfrak{p} \text{ for all } P_1, P_2, P_3 \in \mathfrak{p}.$$

More generally, the spaces $\mathfrak{p}$ and $\mathfrak{k}$ satisfy the following odd-even parity rules under brackets (compare to multiplication table of -1 and 1):

$$[\mathfrak{k}, \mathfrak{k}] \subseteq \mathfrak{k} \tag{2.32}$$
$$[\mathfrak{p}, \mathfrak{k}] \subseteq \mathfrak{p} \tag{2.33}$$
$$[\mathfrak{p}, \mathfrak{p}] \subseteq \mathfrak{k}. \tag{2.34}$$

The decomposition we have introduced is thus closely related to the so called *Cartan decomposition*, see [14]. However, the Cartan decomposition requires σ to be a *Cartan involution*, whose definition involves a certain non-degeneracy of a bilinear form derived from the Cartan-Killing form. For the applications in this paper the Cartan property of σ is *not* needed, thus we have considerable freedom in choosing a suitable σ.

Corresponding to the additive splitting $\mathfrak{g} = \mathfrak{p} \oplus \mathfrak{k}$ there exists a multiplicative splitting of G. Any element $g \in G$ sufficiently close to I can be written as a product

$$g = \exp(P)\exp(K), \text{ where } P \in \mathfrak{p} \text{ and } K \in \mathfrak{k}.$$

The element $\exp(P) \in G$ belongs to a so called *symmetric space*, while $\exp(K)$ belongs to a Lie subgroup of G.

For example, if $G = \mathrm{GL}(n)$ and $\sigma(Z) = -Z^T$, then the splitting $\mathfrak{g} = \mathfrak{p} + \mathfrak{k}$ corresponds to writing a general matrix as the sum of a symmetric and a skew matrix. The corresponding product splitting of G is the *polar decomposition*, where a matrix is written as a product of a symmetric positive definite matrix and an orthogonal matrix. In general, if σ is any involutive automorphism, we will refer to the decomposition as a *generalized polar decomposition*, see [23] for more details.

3 Generalized Polar Coordinates (GPC)

The coordinates we are studying in this paper are recursively defined as follows.

Definition 3.2 GPC denotes any coordinate map from a Lie algebra to a Lie group obtained by the following recursion:

- The exponential map $\exp : \mathfrak{g} \to G$ is GPC on G.
- Let σ be an involutive automorphism on $\mathfrak{g}$ and let $\Pi_\sigma^\pm$ be defined by (2.2) - (2.3). Let $\mathfrak{k} = \text{Range}(\Pi_\sigma^+) \subset \mathfrak{g}$ and $G^\sigma \subset G$ be the corresponding sub-algebra and subgroup. If $\tilde{\Phi} : \mathfrak{k} \to G^\sigma$ is GPC on G^σ then a map $\Phi : \mathfrak{g} \to G$ defined as
$$\Phi(Z) = \exp\left(\Pi_\sigma^- Z\right) \cdot \tilde{\Phi}\left(\Pi_\sigma^+ Z\right), \tag{3.1}$$
is GPC on G.

The coordinates are completely determined by a sequence of involutive automorphisms $\{\sigma_i\}_{i=0}^{k-1}$, giving rise to a sequence of subalgebras $\mathfrak{g} = \mathfrak{g}_0 \supset \mathfrak{g}_1 \supset \cdots \supset \mathfrak{g}_k$, where $\mathfrak{g}_{i+1} = \text{Range}(\Pi_{\sigma_i}^+)$ and $\sigma_i : \mathfrak{g}_i \to \mathfrak{g}_i$. By unfolding the recursion, we obtain the equivalent form of the coordinate map given in (1.1) - (1.2).

Note that we let the -1 eigenspace appear on the left and the +1 on the right in (3.1). This is important if we want to compute right trivialized tangents. If we instead want to work with left trivializations, we must reverse the definition and let the +1 eigenspace appear on the left.

As a trivial example consider $G = \mathbb{C}^*$ (the multiplicative group of nonzero complex numbers), let $\sigma(z) = \bar{z}$ (complex conjugation) and let $\tilde{\Phi}(K) = \exp(K)$. If $Z = X + iY$ then $\Pi_\sigma^- Z = X$, $\Pi_\sigma^+ Z = iY$ and $\Phi(Z) = \exp(X)\exp(iY) = r\exp(i\theta)$, where $r = \exp(X)$ and $\theta = Y$. This yields (classical) polar coordinates on $\mathbb{C}^*$.

3.1 The coordinate map

To obtain efficient algorithms for computing the coordinate map, we assume that σ is a *low rank inner automorphism*, defined as follows.

Definition 3.3 An automorphism $\sigma : \mathfrak{g} \to \mathfrak{g}$ of the form
$$\sigma(Z) = SZS\,, \quad \text{where } S^2 = I \tag{3.2}$$
is called an *inner* automorphism.[1] By the *rank* of an inner automorphism we mean $p = \text{Rank}(\Pi_S^-)$.

If σ is given by (3.2) then $P = \Pi_\sigma^-(Z) = \frac{1}{2}(Z - SZS)$ is 2-cyclic with respect to S, i.e. $SPS = -P$. Theorem 2.2 yields the following formula for computing $\exp(P)$.

Theorem 3.3 *Let $SPS = -P$, where $S^2 = I$. Let $\Theta = P^2\Pi_S^-$, then*
$$\exp(P) = I + \psi_1(\Theta)P + P\psi_1(\Theta) + P\psi_2(\Theta)P + \psi_2(\Theta)\Theta, \tag{3.3}$$

[1] If G is a matrix group, then σ induces the group automorphism $G \ni g \mapsto SgS \in G$. Although it is not necessary that $S \in G$, it must be an element of some larger group containing G as a subgroup, and on this larger group the automorphism is properly of inner type. We stick to the name ‘inner’ also when $S \notin G$.

where

$$\psi_1(x) \quad = \quad \begin{cases} \frac{\sinh(\sqrt{x})}{\sqrt{x}} = \frac{\sin(\sqrt{-x})}{\sqrt{-x}} & \textit{for } x \neq 0 \\ 1 & \textit{for } x = 0 \end{cases} \tag{3.4}$$

$$\psi_2(x) \quad = \quad \begin{cases} 2\frac{\sinh^2(\sqrt{x}/2)}{x} = 2\frac{\sin^2(\sqrt{-x}/2)}{-x} & \textit{for } x \neq 0 \\ \frac{1}{2} & \textit{for } x = 0 \end{cases} \tag{3.5}$$

Proof. From (2.16) we find $\psi_1(x) = (\exp(\sqrt{x}) - \exp(-\sqrt{x}))/(2\sqrt{x}) = \sinh(\sqrt{x})/\sqrt{x}$. Similarly from (2.17) $\psi_2(x) = (\cosh(\sqrt{x}) - 1)/x$, the equivalent form (3.5) being numerically better for small x. □

See notes after Theorem 2.2 for a discussion of the structure of Θ. Combined with the algorithms in Section 2.3 for computing $\psi_i(\Theta)$, we have practical algorithms to be investigated in the sequel.

For the rank 1 case we establish explicit formulae.

Corollary 3.3.1 *Let $SPS = -P$, where $p = \mathrm{Rank}(\Pi_S^-) = 1$. Let the scalar θ be given as $\Pi_S^- P^2 = \theta \Pi_S^-$. Then*

$$\exp(P) = \begin{cases} I + P + \frac{1}{2}P^2 & \textit{if } \theta = 0 \\ I + \frac{\sin(\sqrt{-\theta})}{\sqrt{-\theta}}P + 2\frac{\sin^2(\sqrt{-\theta}/2)}{-\theta}P^2 & \textit{if } \mathbb{R} \ni \theta < 0 \\ I + \frac{\sinh(\sqrt{\theta})}{\sqrt{\theta}}P + 2\frac{\sinh^2(\sqrt{\theta}/2)}{\theta}P^2 & \textit{if } \mathbb{R} \ni \theta > 0 \textit{ or if } \theta \textit{ is complex.} \end{cases} \tag{3.6}$$

It should be noted that this is exactly the same formula as the Rodrigues formula for skew 3×3 matrices. Indeed, let

$$P = \widehat{\boldsymbol{x}} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}$$

and let $S = I - 2\boldsymbol{z}\boldsymbol{z}^T/\boldsymbol{z}^T\boldsymbol{z}$, where $\boldsymbol{z}$ is a vector such that $\boldsymbol{z}^T\boldsymbol{x} = 0$. Then $SPS = -P$. One finds that $\theta = -\boldsymbol{x}^T\boldsymbol{x}$, from which Rodrigues formula follows.

3.2 The tangent map

In this section we will develop the formulae for the computing tangent map of Φ and its inverse. Whereas the theory of the previous section regarded elements of $\mathfrak{g}$ as matrices (linear operators in $\mathbb{R}^{n\times n}$ acting on $\mathbb{R}^n$), we are now concerned with $d\Phi_Z$ and $d\Phi_Z^{-1}$ which belong to the space of all linear operators from $\mathfrak{g}$ to itself, denoted End($\mathfrak{g}$). If $\mathfrak{g}$ is represented as $n \times n$ matrices, then End($\mathfrak{g}$) could be represented as $n^2 \times n^2$ matrices. Inversion of such linear operators by Gaussian elimination would cost $\mathcal{O}(n^6)$ operations, but we are seeking algorithms of complexity at most $\mathcal{O}(n^3)$. To achieve this, we can not rely on a matrix representation of End($\mathfrak{g}$), but rather work directly with operators (projections, involutions) as outlined in Section 2.2.

The theory of this section relies on the odd-even splitting of the subspaces $\mathfrak{g} = \mathfrak{p} \oplus \mathfrak{k}$ in (2.32) - (2.34).

We will develop several formulae for the tangent map of Φ and its inverse. In the following, let σ be an involutive automorphism on $\mathfrak{g}$, and let $P \in \mathfrak{p} \subset \mathfrak{g}$, i.e. $\sigma(P) = -P$. Define the linear operator $\mathrm{ad}_P : \mathfrak{g} \to \mathfrak{g}$ as

$$\mathrm{ad}_P(Z) = [P, Z]. \tag{3.7}$$

Theorem 3.4 *Let $Z = P + K = \Pi_\sigma^- Z + \Pi_\sigma^+ Z$. The right trivialized tangent of Φ and its inverse are given as*

$$
\begin{aligned}
d\Phi_Z &= d\exp_p \Pi_\sigma^- + \mathrm{Ad}_{\exp(P)} d\tilde{\Phi}_K \Pi_\sigma^+ \\
&= \left(\frac{\exp(u) - 1}{u} \Pi_\sigma^- + \exp(u) \Pi_\sigma^+ \right) \left(\Pi_\sigma^- + d\tilde{\Phi}_K \Pi_\sigma^+ \right) \qquad (3.8) \\
{d\Phi_Z}^{-1} &= \left(\Pi_\sigma^- + {d\tilde{\Phi}_K}^{-1} \Pi_\sigma^+ \right) \left(\frac{1 + (u-1)\cosh(u)}{\sinh(u)} \Pi_\sigma^- + (1-u) \Pi_\sigma^+ \right), \qquad (3.9)
\end{aligned}
$$

where $u = \mathrm{ad}_P$.

Proof. The first form of (3.8) follows from

$$
\begin{aligned}
d\Phi_Z(\delta Z)\Phi(Z) &= \left. \frac{\partial}{\partial s} \right|_{s=0} \exp(P + s\delta P) \tilde{\Phi}(K + s\delta K) \\
&= d\exp_P(\delta P) \exp(P) \tilde{\Phi}(K) + \exp(P) d\tilde{\Phi}_K(\delta K) \tilde{\Phi}(K) \\
&= \left(d\exp_P(\delta P) + \exp(P) d\tilde{\Phi}_K(\delta K) \exp(-P) \right) \Phi(Z).
\end{aligned}
$$

Let $u = \mathrm{ad}_P$. Using $d\exp_P = (\exp(u) - 1)/u$, $\mathrm{Ad}_{\exp(P)} = \exp(\mathrm{ad}_P)$, ${\Pi_\sigma^-}^2 = \Pi_\sigma^-$ and $\Pi_\sigma^+ \Pi_\sigma^- = 0$, we obtain the second form of (3.8). To verify (3.9), we observe from (2.32)-(2.34) that if $\psi(x)$ is an analytic function with odd and even parts $\psi(x) = \psi^o(x) + \psi^e(x)$,

$$
\psi^o(x) = \frac{1}{2} \left(\psi(x) - \psi(-x) \right), \; \psi^e(x) = \frac{1}{2} \left(\psi(x) + \psi(-x) \right),
$$

then

$$
\begin{aligned}
\Pi_\sigma^- \psi(u) \Pi_\sigma^- &= \psi^e(u) \Pi_\sigma^-, \; \Pi_\sigma^+ \psi(u) \Pi_\sigma^- = \psi^o(u) \Pi_\sigma^- \\
\Pi_\sigma^- \psi(u) \Pi_\sigma^+ &= \psi^o(u) \Pi_\sigma^+, \; \Pi_\sigma^+ \psi(u) \Pi_\sigma^+ = \psi^e(u) \Pi_\sigma^+.
\end{aligned}
$$

Thus

$$
\begin{aligned}
\Pi_\sigma^- \frac{\exp(u) - 1}{u} \Pi_\sigma^- &= \frac{\sinh(u)}{u} \Pi_\sigma^- \\
\Pi_\sigma^+ \frac{\exp(u) - 1}{u} \Pi_\sigma^- &= \frac{\cosh(u) - 1}{u} \Pi_\sigma^- \\
\Pi_\sigma^- \exp(u) \Pi_\sigma^+ &= \sinh(u) \Pi_\sigma^+ \\
\Pi_\sigma^+ \exp(u) \Pi_\sigma^+ &= \cosh(u) \Pi_\sigma^+.
\end{aligned}
$$

Using these formulae, we find that

$$
\left(\frac{1 + (u-1)\cosh(u)}{\sinh u} \Pi_\sigma^- + (1-u) \Pi_\sigma^+ \right) \left(\frac{\exp(u) - 1}{u} \Pi_\sigma^- + \exp(u) \Pi_\sigma^+ \right) = \Pi_\sigma^- + \Pi_\sigma^+ = I.
$$

Since $\tilde{\Phi}$ is acting only on the subalgebra $\mathfrak{k}$ we have $\Pi_\sigma^- d\tilde{\Phi}_k \Pi_\sigma^+ = 0$ and $\Pi_\sigma^+ d\tilde{\Phi}_k \Pi_\sigma^+ = d\tilde{\Phi}_K \Pi_\sigma^+$ from which we get

$$
\left(\Pi_\sigma^- + {d\tilde{\Phi}_K}^{-1} \Pi_\sigma^+ \right) \left(\Pi_\sigma^- + d\tilde{\Phi}_K \Pi_\sigma^+ \right) = \Pi_\sigma^- + \Pi_\sigma^+ = I,
$$

thus (3.9) is verified. □

Using the same odd-even parity argument as in the proof of Theorem 3.4 we obtain the following result:

Corollary 3.4.1 *The partitioning of $d\Phi_Z$ and $d\Phi_Z{}^{-1}$ with respect to σ is*

$$d\Phi_Z = \begin{pmatrix} \frac{\sinh(u)}{u} & \sinh(u) \\ \frac{\cosh(u)-1}{u} & \cosh(u) \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & d\tilde{\Phi}_K \end{pmatrix} \tag{3.10}$$

$$d\Phi_Z{}^{-1} = \begin{pmatrix} I & 0 \\ 0 & d\tilde{\Phi}_K^{-1} \end{pmatrix} \begin{pmatrix} \frac{u\cosh(u)}{\sinh(u)} & -u \\ \frac{1-\cosh(u)}{\sinh(u)} & 1 \end{pmatrix}, \tag{3.11}$$

where $u = \mathrm{ad}_P$ and $P = \Pi_\sigma^- Z$ restricted to the appropriate subspace.

For efficient computation of the tangent map, it is essential to develop fast algorithms for computing analytic functions of ad_P. First we use the theory of 2-cyclic matrices to simplify (3.9).

Lemma 3.5 *If $\sigma(P) = -P$, then ad_P is 2-cyclic with respect to σ, i.e.*

$$\sigma \mathrm{ad}_P \sigma = -\mathrm{ad}_P. \tag{3.12}$$

Proof. Let $Z \in \mathfrak{g}$ be arbitrary. We have

$$\sigma \mathrm{ad}_P \sigma(Z) = \sigma([P, \sigma(Z)]) = \left[\sigma(P), \sigma^2(Z)\right] = [-P, Z] = -\mathrm{ad}_P(Z).$$

□

Theorem 3.6 *Let σ be an arbitrary involutive automorphism. Let $Z = P + K$ where $P = \Pi_\sigma^- Z$, let $u = \mathrm{ad}_p$ and $\Omega = \mathrm{ad}_P^2 \Pi_\sigma^-$. Then*

$$d\Phi_Z{}^{-1} = \left(\Pi_\sigma^- + d\tilde{\Phi}_K{}^{-1} \Pi_\sigma^+\right) \left(I + u\left(\psi_1(\Omega)\Pi_\sigma^- - \Pi_\sigma^+\right) + \psi_2(\Omega)\Pi_\sigma^-\right), \tag{3.13}$$

where

$$\psi_1(x) = \begin{cases} -\frac{\tanh(\sqrt{x}/2)}{\sqrt{x}} = -\frac{\tan(\sqrt{-x}/2)}{\sqrt{-x}} & \textit{for } x \neq 0 \\ -\frac{1}{2} & \textit{for } x = 0 \end{cases} \tag{3.14}$$

$$\psi_2(x) = \begin{cases} \frac{\sqrt{-x}}{\tanh(\sqrt{-x})} - 1 = \frac{\sqrt{x}}{\tan(\sqrt{x})} - 1 & \textit{for } x \neq 0 \\ 0 & \textit{for } x = 0. \end{cases} \tag{3.15}$$

Proof. Starting from (3.9), we use Lemma 3.5 with Theorem 2.2. Since $u\Pi_\sigma^- = \Pi_\sigma^+ u$, and $\Pi_\sigma^+\Pi_\sigma^- = 0$, all terms of the form $\psi_i(\Omega)\, u\Pi_\sigma^-$ vanish, and the result follows by a straightforward symbolic computation. □

To employ the algorithms of Section 2.3, we must understand the eigenspace structure of Ω. To use Algorithm 2, we use the following result which is an immediate consequence of Theorem 3.9 below. Note that μ_i are the eigenvalues of the $p \times p$ matrix BA in (2.19).

Corollary 3.6.1 *Let $\sigma(P) = SPS = -P$, where $\mathrm{Rank}(\Pi_S^-) = p$. Let $\Theta = P^2 \Pi_S^-$ be nondefect of rank p, with $d \leq p$ different non-zero eigenvalues $\{\mu_i\}_{i=1}^d$. Then $\Omega = \mathrm{ad}_p^2 \Pi_\sigma^-$ is non-defect with $1 + d + d^2$ different eigenvalues given as*

$$0, \quad \{\mu_i\}_i, \quad \left\{\sqrt{\mu_i} - \sqrt{\mu_j}\right\}_{i>j}, \quad \left\{\sqrt{\mu_i} + \sqrt{\mu_j}\right\}_{i \geq j}. \tag{3.16}$$

Thus, the minimal polynomial of Ω is the monic $1 + d + d^2$-degree polynomial with zeros in the points (3.16).

In the rank-1 case, we can obtain explicit formulae. If the scalar θ given by $P^2\Pi_S^- = \theta\Pi_S^-$ is nonzero, then the minimal polynomial for $\Omega = \mathrm{ad}_p^2\Pi_\sigma^-$ becomes

$$q(x) = x(x-\theta)(x-4\theta).$$

From (3.13) we employ Algorithm 2 symbolically to obtain:

Corollary 3.6.2 *Let σ be a rank-1 involutive automorphism. Let $Z = P+K$ where $P = \Pi_\sigma^- Z$, let $u = \mathrm{ad}_p$ and let θ be the scalar $P^2\Pi_S^- = \theta\Pi_S^-$, as in Corollary 3.3.1. If $\theta \neq 0$ then*

$$d\Phi_Z{}^{-1} = \left(\Pi_\sigma^- + d\tilde{\Phi}_K{}^{-1}\Pi_\sigma^+\right)\left(I + u\left((\psi_1(\theta)+\psi_2(\theta)u^2)\Pi_\sigma^- - \Pi_\sigma^+\right) - \left(\psi_3(\theta)u^2 + \psi_4(\theta)u^4\right)\Pi_\sigma^-\right) \tag{3.17}$$

where

$$\begin{aligned}
\psi_1(\theta) &= \frac{-8\tanh(\sqrt{\theta}/2)+\tanh(\sqrt{\theta})}{6\sqrt{\theta}} \\
\psi_2(\theta) &= \frac{2\tanh(\sqrt{\theta}/2)-\tanh(\sqrt{\theta})}{6\theta^{\frac{3}{2}}} \\
\psi_3(\theta) &= \frac{15+\sqrt{\theta}(\tanh(\sqrt{\theta})-15\coth(\sqrt{\theta}))}{12\theta} \\
\psi_4(\theta) &= \frac{-3+\sqrt{\theta}(-\tanh(\sqrt{\theta})+3\coth(\sqrt{\theta}))}{12\theta^2}.
\end{aligned}$$

It should be noted that u^2 and u^4 act on $\mathfrak{p} = \mathrm{Range}\,(\Pi_\sigma^-)$, yielding also a result in $\mathfrak{p}$. If $\mathfrak{g} \subset \mathbb{R}^{n\times n}$ then the cost of computing the terms u^2 and u^4 is $\mathcal{O}(n)$. The main work is the computation of the single u acting on $\mathfrak{p} \oplus \mathfrak{k}$, with a cost of $\mathcal{O}(n^2)$.

Now we return to the general rank p case and the computation of $\psi(\Omega)$ via eigenspace projections. Let P and Ω be as in Lemma 3.6.1. For our present purpose it is not important whether or not μ_i are distinct, so we just assume that Θ is non-defect with p eigenvalues μ_i for $i = 1,\ldots,p$. Let the corresponding left and right eigenvectors be denoted $\boldsymbol{y}_i^T$ and $\boldsymbol{x}_i$, normalized such that

$$\begin{aligned}
\boldsymbol{y}_i^T\Omega &= \mu_i\boldsymbol{y}_i^T \\
\Omega\boldsymbol{x}_i &= \mu_i\boldsymbol{x}_i \\
\boldsymbol{y}_i^T\boldsymbol{x}_j &= \delta_{i,j}.
\end{aligned}$$

The following is verified by straightforward computation.

Lemma 3.7 *P has $2p$ nonzero left and right eigenvectors given as*

$$\boldsymbol{v}_{\pm i} = \frac{1}{\sqrt{2}}\left(\boldsymbol{x}_i \pm P\boldsymbol{x}_i/\sqrt{\mu_i}\right) \quad \text{for } i = 1,\ldots,p \tag{3.18}$$

$$\boldsymbol{w}_{\pm i}^T = \frac{1}{\sqrt{2}}\left(\boldsymbol{y}_i^T \pm \boldsymbol{y}_i^T P/\sqrt{\mu_i}\right) \quad \text{for } i = 1,\ldots,p, \tag{3.19}$$

satisfying

$$\begin{aligned}
P\boldsymbol{v}_{\pm i} &= \pm\sqrt{\mu_i}\boldsymbol{v}_{\pm i} \\
\boldsymbol{w}_{\pm i}^T P &= \pm\sqrt{\mu_i}\boldsymbol{w}_{\pm i}^T \\
\boldsymbol{w}_j^T\boldsymbol{v}_k &= \delta_{j,k} \quad \text{for } j,k \in \{\pm 1,\ldots,\pm p\}.
\end{aligned}$$

Lemma 3.8 *If $\Theta = P^2\Pi_S^-$ non-defect and of rank p, then P is non-defect.*

Proof. In Lemma 3.7 we found $2p$ independent right eigenvectors $\boldsymbol{v}_{\pm i}$. Referring to the partitioned form of P in (2.18), we have that B is a $p \times (n-p)$ matrix with linearly independent rows. Thus there are $n-2p$ zero right eigenvectors of P of the form $\begin{pmatrix} 0 & x^T \end{pmatrix}^T$, where $x \in \text{Ker}(B)$. We have found n independent right eigenvectors of P. □

For $i = 1, \ldots, d$ let the eigenspace projections of P be given as

$$\Pi_{\pm i} = \boldsymbol{v}_{\pm i}\boldsymbol{w}_{\pm i}^T \tag{3.20}$$

$$\Pi_0 = I - \sum_{i=1}^{d} (\Pi_i + \Pi_{-i})\,. \tag{3.21}$$

Now we are ready to formulate the main theorem describing the eigenspace structure of Ω. Note that $\mathfrak{k} \subset \ker(\Omega)$ and $\Omega(\mathfrak{p}) \subset \mathfrak{p}$, thus we can restrict the discussion to the action of Ω on $\mathfrak{p}$.

Theorem 3.9 *Let σ be a rank p inner automorphism, and let $\sigma(P) = -P$. If $\Theta = P^2\Pi_S^-$ is non-defect and of rank p, then $\Omega = \text{ad}_P^2\Pi_\sigma^-$ is non-defect with a complete list of eigenspace projections $\Pi_{i,j} : \mathfrak{p} \to \mathfrak{p}$ expressed in terms of an arbitrary $W \in \mathfrak{p}$ as*

$$\Pi_{0,0}(W) = \Pi_0 W \Pi_0 \tag{3.22}$$

$$\Pi_{i,0}(W) = (\Pi_i + \Pi_{-i})\, W\Pi_0 + \Pi_0 W\,(\Pi_i + \Pi_{-i}) \quad \textit{for } i = 1, \ldots, p \tag{3.23}$$

$$\Pi_{i,j}(W) = \Pi_i W \Pi_j + \Pi_{-i} W \Pi_{-j} \quad \textit{for } i, j = 1, \ldots, p \textit{ and } i > j \tag{3.24}$$

$$\Pi_{i,-j}(W) = \Pi_i W \Pi_{-j} + \Pi_{-i} W \Pi_j \quad \textit{for } i, j = 1, \ldots, p \textit{ and } i \geq j\,. \tag{3.25}$$

The corresponding eigenvalues are given via

$$\Omega\Pi_{0,0}(W) = 0 \tag{3.26}$$

$$\Omega\Pi_{i,0}(W) = \mu_i \Pi_{i,0}(W) \tag{3.27}$$

$$\Omega\Pi_{i,j}(W) = (\sqrt{\mu_i} - \sqrt{\mu_j})^2 \Pi_{i,j}(W) \tag{3.28}$$

$$\Omega\Pi_{i,-j}(W) = (\sqrt{\mu_i} + \sqrt{\mu_j})^2 \Pi_{i,-j}(W). \tag{3.29}$$

Proof. For $\Pi_{i,j}$ we compute

$$\text{ad}_P(\Pi_i W \Pi_j) = P\Pi_i W \Pi_j - \Pi_i W \Pi_j P = (\sqrt{\mu_i} - \sqrt{\mu_j})\Pi_i W \Pi_j.$$

Hence under the action of ad_P^2 the two projections $W \mapsto \Pi_i W \Pi_j$ and $W \mapsto \Pi_{-i} W \Pi_{-j}$ correspond to the same eigenvalue $\left(\sqrt{\mu_i} - \sqrt{\mu_j}\right)^2$. These can be combined into the single projection

$$\Pi_{i,j}(W) = \Pi_i W \Pi_j + \Pi_{-i} W \Pi_{-j}.$$

Now, since $P\Pi_{\pm i} = \pm\sqrt{\mu_i}\Pi_{\pm i}$ and $\sigma(P) = -P$, we find that $\sigma(\Pi_{\pm i}) = \Pi_{\mp i}$. If $W \in \mathfrak{p}$ we have $\sigma(W) = -W$ and hence

$$\sigma(\Pi_{i,j}(W)) = \sigma(\Pi_i W \Pi_j + \Pi_{-i} W \Pi_{-j}) = \Pi_{-i}(-W)\Pi_{-j} + \Pi_i(-W)\Pi_j = -\Pi_{i,j}(W).$$

Thus $W \in \mathfrak{p} \Rightarrow \Pi_{i,j}(W) \in \mathfrak{p}$. We conclude that $\Pi_\sigma^-\Pi_{i,j}(W) = \Pi_{i,j}(W)$, which yields

$$\Omega\Pi_{i,j}(W) = (\sqrt{\mu_i} - \sqrt{\mu_j})^2 \Pi_{i,j}(W).$$

The others follow by a similar computation. To check that we have a complete list of eigenspace projections, we use $\sum_{j=-p}^{p} \Pi_j = I$ to decompose W as

$$W = \left(\sum_{j=-p}^{p} \Pi_j \right) W \left(\sum_{k=-p}^{p} \Pi_k \right),$$

each term $\Pi_j W \Pi_k$ is appearing exactly once in (3.22)-(3.25). □

The computation of these projections should be done with care in order to keep a favourable complexity. It should be noted that for $W \in \mathfrak{p}$ then

$$\Pi_{i,j}(W) = \Pi_i W \Pi_j + \Pi_{-i} W \Pi_{-j} = 2\Pi_\sigma^-(\Pi_i W \Pi_j).$$

If $\mathfrak{g}$ consists of $n \times n$ matrices, then when $i, j \neq 0$ the computation of $\Pi_{i,j}(W)$ is $\mathcal{O}(np)$. The 0-projections Π_0 are typically high rank matrices, and their computation must be done indirectly. After having computed $\Pi_{i,j}(W)$ and $\Pi_{i,-j}(W)$ for non-zero i, j, we find the remaining projections as

$$\begin{aligned}
\tilde{W} &= W - \sum_{i>j} \Pi_{i,j}(W) - \sum_{i \geq j} \Pi_{i,-j}(W) \\
\Pi_{k,0}(W) &= (\Pi_k + \Pi_{-k})\, \tilde{W} \\
\Pi_{0,k}(W) &= \tilde{W}\, (\Pi_k + \Pi_{-k}) \\
\Pi_{0,0}(W) &= \tilde{W} - \sum_{k=1}^{p} \left(\Pi_{k,0}(W) + \Pi_{0,k}(W) \right).
\end{aligned}$$

The total complexity of computing $\psi_i(\Omega)W$ becomes $\mathcal{O}(np^3)$, which is a minor contribution to the total cost as long as $p < \sqrt{n}$.

3.3 A Schur approach

In attempt to overcome possible instabilities if the matrix P becomes close to singular or defect, we will in this section briefly discuss how the Schur decomposition of P can be employed rather than the the eigenvalue decomposition. We start by providing two lemmas concerning analytic functions of the operator ad, and in Section 3.3.2 we briefly discuss how these results apply to computing $\psi_i(\Omega)$ at low complexity.

3.3.1 Computing analytic functions of ad

Let X and Y be arbitrary $n \times n$ matrices. The usual way of describing $\psi(\mathrm{ad}_X)Y$ is by the Taylor expansion of ψ. In the following lemma we give an alternative representation which will prove to be of help in the following.

Lemma 3.10 *Let ψ be an analytic function. Then the following identity holds*

$$\psi(\mathrm{ad}_X)Y = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \psi^{(k)}(X) Y X^k, \tag{3.30}$$

where $\psi^{(k)}$ denotes the k-th derivative of ψ.

Proof. Since ad is a linear operator it suffices to prove the identity for $\psi(x) = x^n$ for $n \in \mathbb{N}$. We have

$$\begin{aligned} \text{ad}_X^n Y = &\ \textstyle\sum_{k=0}^{n} (-1)^k \binom{n}{k} X^{n-k} Y X^k & (3.31) \\ = &\ \textstyle\sum_{k=0}^{n} \frac{(-1)^k}{k!} (\frac{d^k}{dX^k} X^n) Y X^k & (3.32) \\ = &\ \textstyle\sum_{k=0}^{\infty} \frac{(-1)^k}{k!} (\frac{d^k}{dX^k} X^n) Y X^k & (3.33) \end{aligned}$$

which yields the desired result. □

Note that this lemma can be seen as a general form of the well known property $\text{Ad}_{\exp(X)} Y = \exp(\text{ad}_X) Y$. This is seen by setting $\psi = \exp$ in the lemma.

Suppose we have a Schur decomposition $X = QTQ^T$. Then it follows that

$$\psi(\text{ad}_X) Y = Q \left(\psi(\text{ad}_T)(Q^T Y Q) \right) Q^T.$$

Thus we need to investigate the operator ad_T for T upper triangular acting on some general matrix A. The following lemma can be seen as an analog of Parlett's algorithm (see [13]), and can be utilized to compute $\psi(\text{ad}_T) A$

Lemma 3.11 *Let $T = (t_{ij})$ be $n \times n$ upper triangular with distinct eigenvalues, and let A be $n \times n$ arbitrary with columns $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n$. Let ψ be an analytic function. Then each column $\boldsymbol{a}_i^*$ of $\psi(\text{ad}_T A)$ can be obtained recursively as:*

$$\begin{aligned} \boldsymbol{a}_1^* &= \psi(T - t_{1,1} I) \boldsymbol{a}_1 & (3.34) \\ \boldsymbol{a}_i^* &= \psi(T - t_{i,i} I)(\boldsymbol{a}_i - (\boldsymbol{a}_1\ \boldsymbol{a}_2 \cdots \boldsymbol{a}_{i-1})\boldsymbol{x}) + (\boldsymbol{a}_1^*\ \boldsymbol{a}_2^* \cdots \boldsymbol{a}_{i-1}^*)\boldsymbol{x} & (3.35) \end{aligned}$$

where $\boldsymbol{x}$ solves the triangular system

$$\begin{pmatrix} t_{1,1} - t_{i,i} & t_{1,2} & \cdots & t_{1,i-1} \\ & t_{2,2} - t_{i,i} & \cdots & t_{2,i-1} \\ & & \ddots & \vdots \\ & & & t_{i-1,i-1} - t_{i,i} \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} t_{1,i} \\ t_{2,i} \\ \vdots \\ t_{i-1,i} \end{pmatrix}.$$

Proof. Observe that the operator ad is invariant under shifts, thus we have $\text{ad}_T = \text{ad}_{(T - t_{i,i} I)}$. Together with Lemma 3.10 it is therefor clear that for each $i = 1, 2, \cdots, n$, there exist some matrix M such that

$$\begin{aligned} \psi(\text{ad}_T) A &= \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \psi^{(k)}((T - t_{i,i} I)) A (T - t_{i,i} I) \\ &= \psi(T - t_{i,i} I) A - M (T - t_{i,i} I). & (3.36) \end{aligned}$$

In fact, equation (3.36) contains all the information we need to prove the lemma. For $i = 1$, the first column of $(T - t_{1,1} I)$ is zero, and thus we obtain $\boldsymbol{a}_1^* = \psi(T - t_{1,1} I) \boldsymbol{a}_1$. Assuming we know $\boldsymbol{a}_1^*, \boldsymbol{a}_2^*, \cdots, \boldsymbol{a}_{i-1}^*$ we may (since T has distinct eigenvalues) solve equation (3.36) for the first $i-1$ columns of M. Denote these by $\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_{i-1}$. Then column i in the same equation gives

$$\boldsymbol{a}_i^* = \psi(T - t_{i,i} I) \boldsymbol{a}_i - \sum_{k=1}^{i-1} t_{k,i} \boldsymbol{m}_k,$$

which is seen to be equal to (3.35). □

Like in Parlett's algorithm also the above algorithm breaks down if the matrix T has repeated eigenvalues, and a block form of lemma 3.11 is needed. The approach is to reorder and cluster the equal (or sufficiently close) eigenvalues of T in blocks (see [13, 8] for details)

$$T = \begin{pmatrix} T_{1,1} & T_{1,2} & \cdots & T_{1,m} \\ & T_{2,2} & \cdots & T_{2,m} \\ & & \ddots & \vdots \\ & & & T_{m,m} \end{pmatrix}, \tag{3.37}$$

such that $\Lambda(T_{i,i}) = \lambda_i$ and $\Lambda(T_{i,i}) \cap \Lambda(T_{j,j}) = \emptyset,\ i \neq j$. Suppose $T_{i,i}$ is a block of size $m \times m$ with all diagonal values equal to λ. Then $(T_{i,i} - \lambda I)$ is nilpotent, i.e. $(T_{i,i} - \lambda I)^m = 0$, which implies that $(T - \lambda I)^m$ is zero in the place of $T_{i,i}$. By (3.10) it is clear that there exists some matrix M such that

$$\psi(\mathrm{ad}_T)A = \sum_{k=0}^{m-1} \frac{(-1)^k}{k!} \psi^{(k)}(T - \lambda I)A(T - \lambda I)^k - M(T - \lambda I)^m. \tag{3.38}$$

As in the proof of Lemma 3.11 one can now solve for the columns of M corresponding to the *known* columns of $\psi(\mathrm{ad}_T)A$, and since the $T_{i,i}$-block of $(T - \lambda I)^m$ vanishes, the next m columns of $\psi(\mathrm{ad}_T)A$ follows by substitution. (If the $T_{i,i}$-block of $(T - \lambda I)^l$ vanishes for $l < m$, then only the first l terms need to be considered)

It should be noted the method of Lemma 3.11 in general has a complexity of order n^4, and that the block version demands knowledge of the derivatives of ψ up to order equal to the size of the largest block minus one (or even higher if the eigenvalues in a block are only close and not equal). However for our purposes we will typically have p small ($n \gg p$) and thus at most $2p$ nonzero eigenvalues and a big zero block. We comment on this in the next subsection.

3.3.2 Using the Schur decomposition of P

Let $P \in \mathfrak{p}$ be the $n \times n$ matrix represented in the way (2.18) with A and B^T blocks of size $(n-p) \times p$. We wish to find an orthogonal matrix Q and an upper triangular matrix T such that $P = QTQ^T$. Exploiting the special structure of P, this can be obtained as follows:

Step 1: Find Householder matrices $\widehat{Q}_H^{(1)}, \widehat{Q}_H^{(2)}, \cdots, \widehat{Q}_H^{(p)}$ such that $\widehat{Q}_H^T = \widehat{Q}_H^{(p)} \cdots \widehat{Q}_H^{(2)} \widehat{Q}_H^{(1)}$ transforms A to upper triangular. By setting

$$Q_H = \begin{pmatrix} I_p & 0 \\ 0 & \widehat{Q}_H \end{pmatrix},$$

the matrix P is transformed to $\widetilde{P} = Q_H^T P Q_H$ which has its lower $(n - 2p) \times n$-block equal to zero.

Step 2: Find a Schur decomposition of the upper left $2p \times 2p$ block of $\widetilde{P}$, i.e.

$$\widetilde{P}(1 : 2p, 1 : 2p) = \widehat{Q}\widehat{T}\widehat{Q}^T.$$

The Schur vectors of P are now equal to the columns of the matrix

$$Q = Q_H \begin{pmatrix} \widehat{Q} & 0 \\ 0 & I_{n-2p} \end{pmatrix},$$

and the upper triangular matrix $T = Q^T PQ$ is equal to zero in its lower $(n - 2p) \times n$-block. Note that when $n \gg p$, the operation count of the above procedure is of order np^2 (We do of course not form the

Householder matrices explicitly). To apply Lemma 3.11 efficiently we need fast ways to compute functions of T shifted by various values, i.e $\psi(T - \lambda I)$. Denote

$$T = \begin{pmatrix} \widehat{T} & D^T \\ 0 & 0 \end{pmatrix}.$$

By using the commutativity property $\psi(T - \lambda I)T = T\psi(T - \lambda I)$ we obtain

$$\psi(T - \lambda I) = \begin{pmatrix} \psi(\widehat{T} - \lambda I) & \phi_\lambda(\widehat{T})D^T \\ 0 & \psi(-\lambda)I \end{pmatrix}, \tag{3.39}$$

where $\phi_\lambda(x) = (\psi(x-\lambda) - \psi(-\lambda))/x$. Thus the problem reduces to computing functions of $2p \times 2p$ upper triangular matrices. In a recent paper [8], the authors describe a several stage general purpose algorithm for computing matrix functions including a strategy for the reordering and clustering of eigenvalues which was mentioned in the previous subsection. In this way the arithmetical complexity of computing $\psi(T - \lambda I)$ is of order np^2, and since λ takes at most $2p + 1$ different values the overall cost of computing $\psi(\Omega)W$ for some $W \in \mathfrak{p}$ is of order np^3 which again is a minor contribution to the total cost when $n \gg p$.

4 GPCs for matrix groups and symmetric spaces

For the classical matrix groups we can obtain GPCs with the computational complexity $\mathcal{O}(n^3)$ by recursively applying the splitting (3.1). For symmetric spaces our techniques may yield algorithms with optimal complexity, e.g. $\mathcal{O}(n)$ for problems on n-spheres. We illustrate this by some examples.

In Definition 3.2 and the following discussion, we assumed the existence of a nested family of algebras $\mathfrak{g} \supset \mathfrak{g}_1 \supset \cdots \supset \mathfrak{g}_k$ and automorphisms $\sigma_i : \mathfrak{g}_i \to \mathfrak{g}_i$. For matrix algebras $\mathfrak{g} \subset \mathfrak{gl}(n)$ it may be convenient to define σ_i on $\mathfrak{g}$ rather than on $\mathfrak{g}_i$, in which case we must make sure that $\{\sigma_i\}$ define a nested sequence of Lie algebras.

Lemma 4.12 *Let $\mathfrak{g} \subset \mathfrak{gl}(n)$ be a Lie algebra of $n \times n$ matrices. Let $S_i \in \mathfrak{gl}(n)$, $i = 1, \ldots, k$ be involutive matrices such that $\sigma_i(Z) = S_i Z S_i$ are involutive automorphisms on $\mathfrak{g}$ and $\{S_i\}_{i=1}^k$ commute*

$$S_i S_j = S_j S_i \qquad \text{for } i, j \in \{1, \ldots, k\}. \tag{4.1}$$

Then the projections $\Pi^+_{\sigma_i} = \frac{1}{2}(I + \sigma_i)$ define a nested sequence of Lie algebras

$$\mathfrak{g} \supset \mathfrak{g}_1 \supset \cdots \supset \mathfrak{g}_k$$

where $\mathfrak{g}_i = \Pi^+_{\sigma_{i-1}} \Pi^+_{\sigma_{i-2}} \cdots \Pi^+_{\sigma_1}(\mathfrak{g})$ such that $\sigma_i|_{\mathfrak{g}_i}$ are automorophisms on $\mathfrak{g}_i$.

Proof. If $\{S_i\}_i$ commute, then also $\{\Pi_{\sigma_i}\}_i$ commute. Hence $\Pi_{\sigma_i}\Pi_{\sigma_j}\Pi_{\sigma_i} = \Pi_{\sigma_j}\Pi_{\sigma_i}$, and we conclude that $\mathfrak{g}_i \subset \mathfrak{g}_{i-1}$. The rest of the lemma is obvious. □

Important examples of such nested sequences are

$$S_i = I - 2e_i e_i^T,$$

which defines automorphisms for the Lie algebras $\mathfrak{gl}(n)$ and $\mathfrak{so}(n)$, and

$$S_i = I - 2(e_i e_i^T + e_{i+m} e_{i+m}^T),$$

which yields automorphisms on the symplectic algebra $\mathfrak{sp}(2m)$.

4.1 Low rank GPC for matrix Lie groups

In this section we give explicit formulae for the tangent maps of some low rank generalized polar coordinate maps, and comment on their arithmetical complexity. For details on the implementation of the coordinate map we refer to [27]. The simplest approach is to use rank-1 splittings, in particular the one resulting from the inner automorphisms $\sigma_i(M) = S_i M S_i$ with $S_i = I - 2e_i e_i^T$. We thus obtain a splitting of $Z \in \mathfrak{g}$:

$$Z = P_1 + P_2 + \cdots + P_{n-1} + K_{n-1},$$

where for $K_0 := Z$ and for $i = 1, \cdots, n-1$ we have

$$P_i = \Pi^-_{\sigma_i}(K_{i-1}) \quad \text{and} \quad K_i = \Pi^+_{\sigma_i}(K_{i-1}).$$

We thus use the map

$$\Phi(Z) = \exp(P_1)\exp(P_2)\cdots\exp(P_{n-1})\exp(K_{n-1}). \tag{4.2}$$

Note however that this approach does not work for every Lie group, i.e. one needs to make sure that the summands P_i and K_i both lie in the Lie algebra, such that the coordinate map $\Phi(Z)$ resides in the Lie group. As we shall see in the sequel, the symplectic group requires a rank two splitting in order for the coordinate map to reside in the Lie group.

For simplicity we consider the first step in the recursive splitting above. Let $Z, \widehat{Z} \in \mathfrak{g}$, and denote their splittings $Z = P + K$ and $\widehat{Z} = \widehat{P} + \widehat{K}$, and further

$$P = \begin{pmatrix} 0 & \boldsymbol{b}^T \\ \boldsymbol{a} & 0 \end{pmatrix} \qquad \widehat{P} = \begin{pmatrix} 0 & \boldsymbol{d}^T \\ \boldsymbol{c} & 0 \end{pmatrix}. \tag{4.3}$$

Then by Theorem 3.4, for the coordinate map $\Phi(Z) = \exp(P)\widetilde{\Phi}(K)$, its inverse tangent applied to $\widehat{Z}$, splits as follows:

$$\begin{aligned} \mathfrak{p} - \text{part} \quad &: \quad (I + \psi_2(\mathrm{ad}_P^2))\widehat{P} - \mathrm{ad}_P \widehat{K} & (4.4)\\ \mathfrak{k} - \text{part} \quad &: \quad d\widetilde{\Phi}_K^{-1}(\widehat{K} + \mathrm{ad}_P(\psi_1(\mathrm{ad}_P^2)\widehat{P})), & (4.5) \end{aligned}$$

where ψ_1 and ψ_2 are given in (3.14)-(3.15). Using the eigenspace projections of Theorem 3.9 for $\Omega = \mathrm{ad}_P^2 \Pi_\sigma^-$, we obtain the following expression for $\psi_i(\mathrm{ad}_P^2)\widehat{P}$:

$$\psi_i(\mathrm{ad}_P^2)\widehat{P} = \begin{pmatrix} 0 & \boldsymbol{y}^T \\ \boldsymbol{x} & 0 \end{pmatrix} \tag{4.6}$$

where

$$\begin{aligned} \boldsymbol{x} &= \psi_i(\theta)\boldsymbol{c} + (-2\phi_i(4\theta)(\boldsymbol{b}^T\boldsymbol{c} - \boldsymbol{d}^T\boldsymbol{a}) - \phi_i(\theta)\boldsymbol{b}^T\boldsymbol{c})\boldsymbol{a} & (4.7)\\ \boldsymbol{y} &= \psi_i(\theta)\boldsymbol{d} + (2\phi_i(4\theta)(\boldsymbol{b}^T\boldsymbol{c} - \boldsymbol{d}^T\boldsymbol{a}) - \phi_i(\theta)\boldsymbol{d}^T\boldsymbol{a})\boldsymbol{b}. & (4.8) \end{aligned}$$

Here $\theta = \boldsymbol{b}^T\boldsymbol{a}$ and ϕ_i, $i = 1, 2$ are the functions $\phi_i(x) = (\psi_i(x) - \psi_i(0))/x$. ($\phi_i(0) = \psi_i'(0)$ is defined since ψ_i is analytic.) Letting $[\boldsymbol{x}, \boldsymbol{y}] = \mathrm{fad2}(\psi, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d})$ denote the function giving the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ above, we get the following algorithm for the tangent map (in MATLAB syntax).

Algorithm 3

```
% In:  n × n matrices Z and Ẑ
% Out:  Ẑ overwritten as dΦ_Z^{-1} Ẑ
```

```
for k = 1 : n − 1
    a := Z(k + 1 : n, k);     b := Z(k, k + 1 : n)^T;
    c := Ẑ(k + 1 : n, k);     d := Ẑ(k, k + 1 : n)^T;
    [x_1, y_1] :=fad2(ψ_1, a, b, c, d);     [x_2, y_2] :=fad2(ψ_2, a, b, c, d);
    Ẑ(k + 1 : n, k) := c + x_2 − aẐ(k, k) + Ẑ(k + 1 : n, k + 1 : n)a;
    Ẑ(k, k + 1 : n) := d^T + y_2^T − b^T Ẑ(k + 1 : n, k + 1 : n) + Ẑ(k, k)b^T;
    Ẑ(k, k) := Ẑ(k, k) + b^T x_1 − y_1^T a;
    Ẑ(k + 1 : n, k + 1 : n) := Ẑ(k + 1 : n, k + 1 : n) + a y_1^T − x_1 b^T;
end
```

4.1.1 The general linear group and the special linear group

Recall that the general linear group $GL(n)$ consists of the $n \times n$ matrices with nonzero determinant, and that the special linear group $SL(n)$ consists of the matrices with determinant equal to one. Their corresponding Lie algebras $\mathfrak{gl}(n)$ and $\mathfrak{sl}(n)$ consist of the $n \times n$ matrices and the $n \times n$ matrices with zero trace respectively. For both Lie algebras it is easy to see that for a rank one splitting $Z = P + K$ with S as above, both P and K lie in the Lie algebra, and thus also the coordinate map resides in the Lie group. By studying Algorithm 3 one sees that at stage k, we perform two matrix vector products, two outer products of vectors and two matrix additions which amounts to a complexity of $8(n-k)^2$. Moreover the operation count of order $n-k$ can with a small effort at least be reduced to 19. Summing from $k = 1$ to n, the overall cost of computing the tangent map is $\frac{8}{3}n^3 + \frac{11}{2}n^2 + \mathcal{O}(n)$.

4.1.2 The orthogonal group

For the orthogonal group $O(n)$, and its corresponding Lie algebra $\mathfrak{so}(n)$, the formulas above simplifies considerably. Let P and $\widehat{P}$ be the matrices (4.3), but with $\boldsymbol{b}$ and $\boldsymbol{d}$ replaced with $-\boldsymbol{a}$ and $-\boldsymbol{c}$ respectively. Then $\psi_i(\mathrm{ad}_P^2)\widehat{P}$ have the form:

$$\psi_i(\mathrm{ad}_P^2)\widehat{P} = \begin{pmatrix} 0 & -\boldsymbol{x}^T \\ \boldsymbol{x} & 0 \end{pmatrix} \tag{4.9}$$

with

$$\boldsymbol{x} = \psi_i(\theta)\boldsymbol{c} + \phi_i(\theta)(\boldsymbol{a}^T\boldsymbol{c})\boldsymbol{a}. \tag{4.10}$$

Also the overall algorithm simplifies. By exploiting the skew symmetry of the output matrix, one sees that at stage k one needs one matrix vector product, one vector outer product, and two half matrix additions, i.e. $4(n-k)^2$ operations. The $(n-k)$ coefficient can be reduced to 8, and summing up we achieve the computational cost $\frac{4}{3}n^3 + 2n^2 + \mathcal{O}(n)$.

4.1.3 Upper triangular group

We also include the upper triangular group in this discussion, all though the formulas turn out rather trivial. By setting the vectors $\boldsymbol{a}$ and $\boldsymbol{c}$ in the matrices (4.3) equal to zero, it is clear that $\mathrm{ad}_P\widehat{P} = 0$, and thus $\psi_i(\mathrm{ad}_P)\widehat{P} = \widehat{P}$. At step k in the algorithm there is only one triangular matrix vector product which may be carried out in $(n-k)^2$ operations. The rest of the computation requires about $3(n-k)$ operations, and overall we end up with cost $\frac{1}{3}n^3 + \frac{1}{2}n^2 + \mathcal{O}(n)$.

4.1.4 The symplectic group

Recall that the symplectic group $\mathrm{SP}(2m)$ are the matrices X satisfying the property

$$XJX^T = J, \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}.$$

Its Lie algebra $\mathfrak{sp}(2m)$ is characterized by the matrices Z which obeys $ZJ + JZ^T = 0$. Note that a rank-1 splitting of the matrix Z will not fulfill this property, i.e. for $Z = P + K$ we will in general have $PJ - JP \neq 0$. Instead consider the involutive $S = I - 2e_1e_1^T - 2e_{m+1}e_{m+1}^T$. It is easily seen that the splitting $Z = P + K$ according to S has the desired property, and that by applying suitable permutation matrices Π (exchanging row/column 2 and $m+1$), the matrix P has the form

$$P = \Pi \begin{pmatrix} 0 & B^T \\ A & 0 \end{pmatrix} \Pi,$$

where A and B are $(2m-2) \times 2$ matrices. Note that because of the symmetries in P, the matrix B is completely determined by A. Furthermore one sees that B^TA is just a scalar times the 2×2 identity matrix, i.e. $B^TA = \theta I$, and thus $\psi(\mathrm{ad}_P)\widehat{P}$ is given

$$\psi(\mathrm{ad}_P^2)\widehat{P} = \Pi \begin{pmatrix} 0 & Y^T \\ X & 0 \end{pmatrix} \Pi, \tag{4.11}$$

where

$$X = \quad \psi_i(\theta)C + A(-2\phi_i(4\theta)(B^TC - D^TA) - \phi_i(\theta)B^TC) \tag{4.12}$$

$$Y = \quad \psi_i(\theta)D + B(2\phi_i(4\theta)(B^TC - D^TA) - \phi_i(\theta)D^TA). \tag{4.13}$$

Omitting the details we can also in this case exploit the symmetry of the matrices involved, and obtain the same computational cost for the tangent map as for the orthogonal group, i.e. $\frac{4}{3}n^3 + \mathcal{O}(n^2)$ for $n = 2m$.

Summarizing we obtain the following table for the leading term of the computational complexity for the maps Ψ and $d\Psi_Z^{-1}$ applied to the most common real matrix Lie algebras.

	Tangentmap	Coordinatemap	
		vector	matrix
$\mathfrak{sl}(n)$, $\mathfrak{gl}(n)$	$\frac{8}{3}n^3$	$3n^2$	$2n^3$
$\mathfrak{o}(n)$	$\frac{4}{3}n^3$	$3n^2$	$2n^3$
$\mathfrak{tu}(n)$	$\frac{1}{3}n^3$	-	$\frac{1}{2}n^3$
$\mathfrak{sp}(n)$	$\frac{4}{3}n^3$	-	$2n^3$

4.2 Symmetric spaces

Symmetric spaces is an important class of manifolds of which n-spheres constitute the most well known example. Also the space of all symmetric matrices as well as the Grassman manifolds are examples of symmetric spaces.

Definition 4.4 A symmetric space is a Riemannian manifold $\mathcal{M}$ with the property that for any point $p \in \mathcal{M}$ there exists an isometry S_p of $\mathcal{M}$ for which p is an isolated fix point.

Let G be the Lie group of all isometries of $\mathcal{M}$ and $\mathfrak{g}$ its Lie algebra. Given a point $p \in \mathcal{M}$, we define an inner involutive automorphism on $\mathfrak{g}$ via

$$\sigma_p(Z) = S_p Z S_p.$$

Consider a timestep with Algorithm 1 based on the GPC obtained from σ_{y_n}. Every movement on $\mathcal{M}$ is computed as

$$\Phi(Z)\cdot y_n = \exp(P)\exp(K)\cdot y_n,$$

where

$$Z = P + K, \qquad \sigma_{y_n}(P) = S_{y_n} P S_{y_n} = -P, \qquad \sigma_{y_n}(K) = S_{y_n} K S_{y_n} = K.$$

Important computational savings arise from the following result:

Lemma 4.13 *Let $K \in \mathfrak{g}$ be such that $\sigma_{y_n}(K) = K$. Then*

$$\exp(K)\cdot y_n = y_n.$$

Proof. Since S_{y_n} is involutive we find

$$S_{y_n}(\exp(K)\cdot y_n) = S_{y_n}\exp(K) S_{y_n} S_{y_n}\cdot y_n = \exp(S_{y_n} K S_{y_n})\cdot y_n = \exp(K)\cdot y_n,$$

hence $\exp(K)\cdot y_n$ is a fixpoint of σ_{y_n}. Assume that $\exp(K)\cdot y_n \neq y_n$. Since $\exp(K) = \exp(K/j)^j$, we find that $\exp(K/j)\cdot y_n$ are fixpoints of S_{y_n} arbitrarily close to y_n. This contradicts the assumption that y_n is isolated, and we conclude that

$$\exp(K)\cdot y_n = y_n.$$

□

In fact, σ_{y_n} yields a splitting $\mathfrak{g} = \mathfrak{p} \oplus \mathfrak{k}$ where $\mathfrak{p}$ is the *canonical horizontal bundle* and $\mathfrak{k}$ is the *normal bundle* at y_n. Symmetric spaces differ from homogeneous manifolds by the fact that symmetric spaces possess a canonical projection onto the horizontal bundle, given by $\Pi^-_{\sigma_{y_n}} = \frac{1}{2}(I - \sigma_{y_n})$.

In the rest of this section we concentrate on the important example of *spherical integrators*, for equations evolving on the unit ball in a Hilbert space. Given a Hilbert space $\mathcal{H}$ with an inner product $\langle x, y\rangle$. Let y^T denote the dual with respect to the given inner product and $||y|| = \langle y, y\rangle^{\frac{1}{2}}$. Given a differential equation

$$y'(t) = A(y)\cdot y, \qquad ||y(0)|| = 1, \tag{4.14}$$

where $y(t) \in \mathcal{H}$ and $A(y)$ is a skew linear operator on $\mathcal{H}$, $\langle u, A(y)\cdot v\rangle = -\langle A(y)\cdot u, v\rangle$. By differentiation we find that $\frac{\mathrm{d}}{\mathrm{d}t}||y(t)|| = 0$, hence the equation evolves on the unit sphere $\mathcal{S} = \{\, y \in \mathcal{H} \mid ||y|| = 1 \,\}$.

$\mathcal{S}$ is a symmetric space where the isometry S_{y_n} is given as

$$S_{y_n} = 2 y_n y_n^T - I. \tag{4.15}$$

As an example of (4.14), take $\mathcal{H}$ as $\mathbb{R}^n$ with the standard inner product, let $A(y) \in \mathbb{R}^{n\times n}$ be a skew-symmetric matrix and $A(y)\cdot v$ be standard matrix-vector product. Our numerical integrator discussed below is based solely on computing the tangent vector $t = A(y)\cdot v$ rather than forming $A(y)$ explicitly. For many computations this can be computed in $\mathcal{O}(n)$ flops, yielding algorithms with optimal complexity $\mathcal{O}(n)$ per step.

Other interesting examples are PDEs where the L^2 norm of the solution is preserved, e.g. Schrödinger equations and the KdV equation. For example, take $\mathcal{H} = L^2([0, 2\pi])$ as the space of 2π periodic functions with the standard integral L^2 inner product. Let $A(y)$ be the (skew symmetric) differential operator

$$A(y) = -\frac{1}{3}\left(y\partial_x + \partial_x y\cdot\right) - \partial_x^3,$$

such that

$$A(y)\cdot v = -\frac{1}{3}\left(yv_x + (yv)_x\right) - v_{xxx}.$$

Inserted in (4.14) this yields the KdV

$$y_t = -yy_x - y_{xxx}.$$

If $A(y)$ is discretized with finite differences, then computation of $A(y)\cdot v$ costs $\mathcal{O}(n)$ flops and with spectral discretizations $\mathcal{O}(n\log(n))$ flops.

We return to the development of spherical integrators of the form Algorithm 1, based on the GPC derived from the canonical horizontal projection at y_n,

$$\Phi(Z)\cdot y_n = \exp(\Pi^-_{\sigma_{y_n}} Z)\cdot y_n. \tag{4.16}$$

Note that in Algorithm 1 we only need to compute the horizontal $\mathfrak{p}$-components of U_i, K_i and $\tilde{K}_i$. This subspace is characterized by:

Lemma 4.14 *Let Z be a skew operator. Then*

$$P = \Pi^-_{\sigma_{y_n}} Z = vy_n^T - y_n v^T, \qquad \textit{where } v = Z\cdot y_n. \tag{4.17}$$

Proof. Check that if P is given by (4.17) then

$$\begin{aligned} \langle v, y_n\rangle &= 0 \\ Py_n &= Zy_n \\ \sigma_{y_n}(P) &= -P, \end{aligned}$$

hence P is the horizontal component of Z. □

Hence $\mathfrak{p}$ is identified with the subspace of $\mathcal{H}$ consisting of all vectors $v \in \mathcal{H}$ orthogonal to y_n. This result might not come as a big surprise, but is crucial for the complexity of computations!

The computation of motions on the sphere S is simplified by the following result:

Lemma 4.15 *Let Z, P, y_n and v be as above. Then*

$$\Phi(Z)\cdot y_n = \exp(\Pi^-_{\sigma_{y_n}} Z)\cdot y_n = \cos(||v||)y_n + \frac{\sin(||v||)}{||v||}v. \tag{4.18}$$

Proof. Check that $Py_n = v$ and $P^2 y_n = -||v||^2 y_n$. The result follows from (3.6), where $\theta = -||v||^2$. □

The computation of inverse tangent maps is simplified by the following result:

Lemma 4.16 *For an arbitrary $Z \in \mathfrak{g}$ let $P = \Pi^-_{\sigma_{y_n}} Z = vy_n^T - y_n v^T$. Then*

$$\Pi^-_{\sigma_{y_n}} d\Phi_Z^{-1} = \Pi^-_{\sigma_{y_n}} + \frac{\tan(||v||) - ||v||}{||v||^2}\mathrm{ad}_P^2 \Pi^-_{\sigma_{y_n}} - \Pi^-_{\sigma_{y_n}}\mathrm{ad}_P. \tag{4.19}$$

Proof. Applying $\Pi^-_{\sigma_{y_n}}$ on (3.17) yields

$$\Pi^-_{\sigma_{y_n}} d\Phi_Z^{-1} = \Pi^-_{\sigma_{y_n}} - \Pi^-_{\sigma_{y_n}} u - \Pi^-_{\sigma_{y_n}} \left(\psi_3(\theta)u^2 + \psi_4(\theta)u^4\right),$$

where $u = \mathrm{ad}_P$ and $\theta = -||v||^2$. For operators of the form (4.17) a direct computation shows that $\Pi^-_{\sigma_{y_n}} u^4 = \theta \Pi^-_{\sigma_{y_n}} u^2$, and the result follows by trigonometric manipulation. □

To complete the computation of the inverse tangent, we need the following result, which follows from direct computation, using skew symmetry and the fact that $\Pi^-_{\sigma_{y_n}} \mathrm{ad}_P^2 = \mathrm{ad}_P^2 \Pi^-_{\sigma_{y_n}}$:

Lemma 4.17 *Let P and v be as above and W an arbitrary skew operator. Then*

$$\Pi^-_{\sigma_{y_n}}[P, W] = w y_n^T - y_n w^T, \qquad w = \langle W\cdot v, y_n\rangle y_n - W\cdot v \tag{4.20}$$

and

$$\Pi^-_{\sigma_{y_n}}[P, [P, W]] = q y_n^T - y_n q^T, \qquad q = -\langle v, v\rangle W\cdot y_n + \langle v, W\cdot y_n\rangle v. \tag{4.21}$$

We conclude with a count of the cost for the coordinate map and the inverse tangent in the case of the n-sphere:

$$\begin{aligned} \Phi(P)\cdot y_n \quad &: \quad 5n \text{ flops} \\ \Pi^-_{\sigma_{y_n}} d\Phi_Z^{-1}(W) \quad &: \quad 10n \text{ flops } + 2 \text{ eval. of } W\cdot y. \end{aligned}$$

5 Numerical experiments

5.1 Finding a solution of the Schur-Horn majorization problem

As an application, we consider an inverse eigenvalue problem with prescribed entries along the main diagonal [6] which arises in conjunction with the Schur-Horn theorem in linear algebra. Before presenting the theorem, we recall that a vector $\mathbf{a} \in \mathbb{R}^n$ is said to *majorize* $\boldsymbol{\lambda} \in \mathbb{R}^n$ if, assuming the ordering

$$\begin{aligned} a_{j_1} \le a_{j_2} \le \cdots \le a_{j_n}, \\ \lambda_{m_1} \le \lambda_{m_2} \le \cdots \le \lambda_{m_n}, \end{aligned}$$

the following relationship holds:

$$\sum_{i=1}^{k} \lambda_{m_i} \le \sum_{i=1}^{k} a_{j_i}, \qquad \text{for } k = 1, 2, \ldots, n, \tag{5.1}$$

and with equality for $k = n$ [15].

Theorem 5.18 (Schur-Horn, [15]) *An Hermitian matrix H with eigenvalues $\boldsymbol{\lambda}$ and diagonal elements $\mathbf{a}$ exists if and only if $\mathbf{a}$ majorizes $\boldsymbol{\lambda}$. Moreover, if $\mathbf{a}$ majorizes $\boldsymbol{\lambda}$ the matrix H can be chosen to be symmetric.*

Given $\boldsymbol{\lambda}$ and $\mathbf{a}$ majoirizing $\boldsymbol{\lambda}$, we wish to find such a symmetric matrix H. One possibility is to construct an isospectral flow (a matrix flow which leaves the eigenvalues of the initial matrix unchanged) which converges to a target symmetric matrix H, similar to diag($\boldsymbol{\lambda}$) and with $\mathbf{a}$ as diagonal elements. Setting $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$, on the isospectral manifold $\mathcal{M}_{\boldsymbol{\lambda}} = \{X : X = U\Lambda U^T, UU^T = I\}$ of symmetric matrices

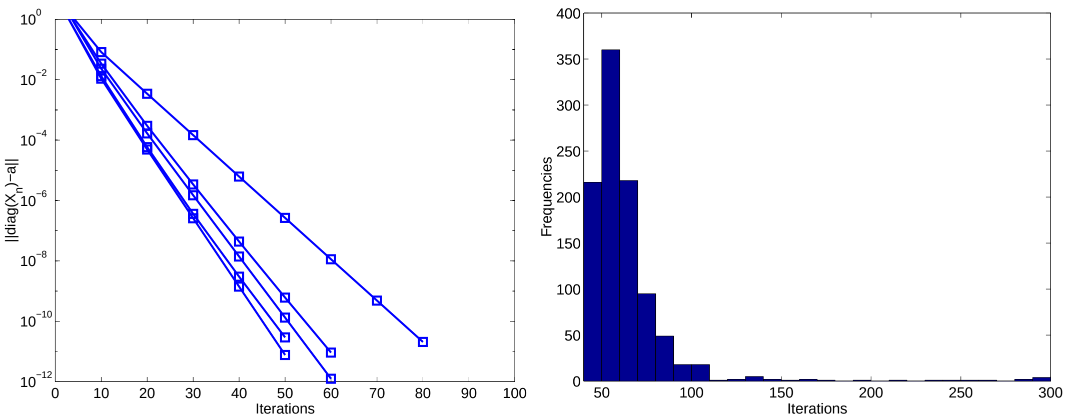

Figure 1: Convergence of X_n for five randomly chosen initial data $\boldsymbol{\lambda}, \boldsymbol{a} \in \mathbb{R}^{25}$ (left), and histogram for the same example repeated 1000 times (right).

with eigenvalues $\boldsymbol{\lambda}$ the problem can be reformulated as an optimization problem: find a matrix H that minimizes the quadratic function

$$\phi(X) = (x_{1,1} - a_{j_1})^2 + \cdots + (x_{n,n} - a_{j_n})^2. \tag{5.2}$$

Following [26], we consider ϕ as a function of U and notice that

$$\phi(U) = \mathrm{tr}[(X - A)\mathrm{diag}(X - A)],$$

where, for convenience, we have set $A = \mathrm{diag}(\mathbf{a})$. By direct computation,

$$\nabla \phi_U = 2[X, \mathrm{diag}(X - A)]U,$$

(here the square bracket denotes the usual matrix commutator, which leads to the double-bracket isospectral flow

$$X' = 2[[X, \mathrm{diag}(X - A)], X]. \tag{5.3}$$

It is not a good idea to solve (5.3) numerically directly, since it is well known that standard ODE methods cannot preserve isospectrality [2]. Instead, assuming that $X_n \in \mathcal{M}_{\boldsymbol{\lambda}}$ is known, in each interval $[t_n, t_{n+1}]$ one solves for the matrix U, obeying the Lie-group differential equation

$$U' = 2[X, \mathrm{diag}(X - A)]U, \qquad U(t_n) = I, \tag{5.4}$$

in tandem with the transformation $X = U(t)X_nU(t)^T$. As long as the solution of (5.4) is orthogonal (or skew-Hermitian, in the complex setting), the numerical approximation X_n has eigenvalues $\boldsymbol{\lambda}$ and it converges to a solution H minimizing (5.2).

In the numerical experiments we chose $\boldsymbol{\lambda}$ and $\boldsymbol{a}$ from random 25×25 symmetric matrices with distribution $\mathcal{N}(0, 1)$ obtained from the MATLAB function `randn`. The initial value X_0, is set to $X_0 = Q^T \Lambda Q$ where Q is a randomly chosen orthogonal matrix. Since we are interested in convergence to a fixed point of (5.4), the local error is not of concern, and thus a first order method works as well as a higher order method. We have used the Lie-Euler method with constant stepsize $h = 0.015$. We use the GPC coordinate map (4.2) and also the matrix exponential for comparisons. Running the two methods on a set $\{\boldsymbol{\lambda}, \boldsymbol{a}\}_i$, $i = 1, 2, \cdots, 1000$, we found that the difference in rate of convergence for using the GPC map and the exponential map was negligible. In Figure 5.1 on the right the histogram plot for the 1000 trials are presented. Among the trials there were two cases outside the region of the plot, with a maximum number of 550 iterations until convergence. In this particular case with 25×25 matrices the evaluation of the exponential map applied to a matrix requires more than six times the number of operations required for the GPC map, and thus considerable savings are obtained.

5.2 Computing all Lyapunov exponents for a ring of oscillators

As a numerical example where the tangent map is needed, we consider computing all Lyapunov exponents of the following system

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

It describes a ring of n damped oscillators with amplitudes x_i with periodic boundary conditions $x_{n+1} = x_1$. The ring is forced externally by $y(t)$, the periodic space coordinate of the limit cycle of a van der Pol oscillator. The parameters $\alpha, \beta, \omega, \sigma$ and d_i are chosen as in [1] to obtain several positive exponents, i.e $\alpha = 1$, $\beta = 1$, $\omega = 1.82$ 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 potential function V is given $V(x) = x^2/2 + x^4/4$. The experiment is done with $n = 5$. The Lyapunov exponents give the rates of exponential divergence or convergence of initial nearby orbits, and can be found by considering the system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ linearized about a trajectory $\boldsymbol{x}(t)$:

$$\dot{Y} = A(t)Y, \quad Y(0) = Y_0, \tag{5.5}$$

where $A(t) = df(\boldsymbol{x}(t))$. The Lyapunov exponents are now given as the logarithms of the eigenvalues of the *Oseledec*-matrix

$$\Lambda_{\boldsymbol{x}} = \lim_{t\to\infty}(Y(t)^T Y(t))^{\frac{1}{2t}}. \tag{5.6}$$

A technique using *continuous QR-factorization* [10, 9] attempts to calculate the orthonormal factor $Q(t)$ in the QR-decomposition of $Y(t)$. It can be shown that $Q(t)$ obeys the (Lie group) differential equation

$$\dot{Q} = QH(t,Q), \quad H(t,Q) = \mathrm{tril}(Q^T AQ) - \mathrm{tril}(Q^T AQ)^T. \tag{5.7}$$

Here $\mathrm{tril}(M)$ denotes the function setting the upper triangular part of M to zero. Since the columns of $Q(t)$ are drawn towards the direction of the largest Lyapunov exponents it is crucial that the numerical solution stays orthonormal, and using standard methods the numerical solution will typically blow up after some time. Thus it is a good idea to use methods which conserves the orthogonality automatically. Given $Q(t)$, it can be shown that the Lyapunov exponents can be obtained from the diagonal elements of the limit matrix

$$\lim_{t\to\infty}\frac{1}{t}\int_0^t Q(\tau)^T A(\tau)Q(\tau)d\tau. \tag{5.8}$$

In numerical computations it is necessary to truncate the above expression at some finite time T, to obtain approximations to the exponents. We have used the trapezoidal rule to approximate the integral (5.8).

Since the tangent vectors in (5.7) are represented by an element in the Lie algebra multiplied from the left rather than the right, we use a *left* version of the coordinate map and the corresponding *left trivialized* tangent map. Letting Φ be the map (4.2), we define for $Z \in \mathfrak{so}(n)$

$$\begin{aligned}
\Phi^*(Z) &= \Phi(-Z)^{-1} \\
&= \exp(P_{n-1})\cdots\exp(P_1).
\end{aligned} \tag{5.9}$$

The left trivialized tangent $d\Phi^*_Z$ of the map Φ^* is then given by the relation

$$\begin{aligned}
\Phi^*(Z)d\Phi^*_Z(\delta Z) &= \left.\frac{\partial}{\partial s}\right|_{s=0}\Phi^*(Z+s\delta Z) \\
&= \left.\frac{\partial}{\partial s}\right|_{s=0}\Phi(-Z-s\delta Z)^{-1} && (5.10)\\
&= -\Phi(-Z)^{-1}d\Phi_{-Z}(-\delta Z)\Phi(-Z)\Phi(-Z)^{-1} && (5.11)\\
&= \Phi^*(Z)d\Phi_{-Z}(\delta Z),
\end{aligned}$$

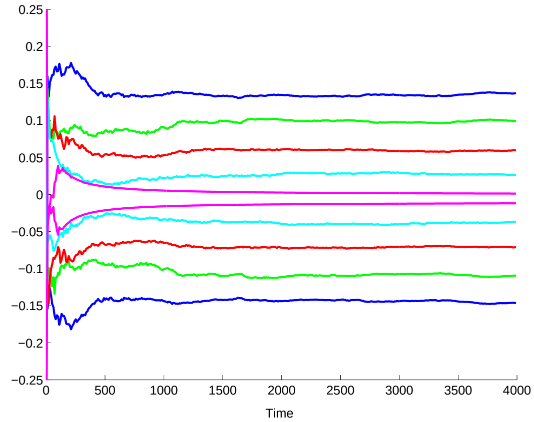

Figure 2: Lyapunov exponents for a ring of 5 damped oscillators.

where in step (5.10)-(5.11) we have used the relation $\frac{d}{dt}(M^{-1}) = -M^{-1}(\frac{d}{dt}M)M^{-1}$. Thus the left trivialized tangent of the map Φ^* is simply given as $d\Phi^*_Z = d\Phi_{-Z}$. Also by noting that $\mathrm{ad}^2_{-P} = \mathrm{ad}^2_P$, we see that $d\Phi^{*\,-1}_Z$ is given by the formula in Theorem 3.6 with just a single sign change in front of u. This is similar to the left trivialized tangent of the exponential map given as $d\exp_{-Z}$.

We use the classical fourth order Runge-Kutta method both for the computation of the trajectory and as the underlying method for the Lie group integrator for solving (5.7) with step sizes $h = 0.005$ and $h = 0.01$ respectively. A randomly chosen $\boldsymbol{x}_0$ is used as initial value, and the system is integrated from $t = 0$ to $t = 4000$. For our choice of damping parameters d_i, the sum of the exponents should add up to

$$\sum_{k=1}^{2n} \lambda_k = \mathrm{div} f(\boldsymbol{x}) = \sum_{j=1}^{n} d_j.$$

In the numerical computations the error of the sum is of order 10^{-8}. Moreover it is shown in [11] that for constant d, the exponents are distributed symmetrically around $-d$. In our case one can also show that the exponents come in pairs $(\lambda_i, -\lambda_i - c_i)$ $i = 1, \cdots, n$, where each c_i satisfies $d_{\min} \leq c_i \leq d_{\max}$. This property is clearly seen in Figure 5.2.

We also performed the experiment using the matrix exponential and its left trivialized tangent giving similar qualitative results. However the cost of the overall algorithm increased dramatically. Using `flops` in MATLAB, the overall cost when using the GPC-map was 1.60×10^{10} while for the exponential map 5.70×10^{10}. For comparison we also implemented the Cayley map (see [16]), and obtained the flop count 2.45×10^{10}.

In this example we computed all exponents of our system. There has been a lot of work concerning computing the few largest Lyapunov exponents of dynamical systems [10]. This is possible by considering a more complicated form of the equation (5.7) on the Stiefel manifold. In [19] the GPC approach is adapted to equations on the Stiefel manifold in such a manner that favorable complexity is achieved.

6 Concluding remarks

We have presented a general theory of splitting methods for obtaining coordinates on Lie groups, where both the coordinate maps and the tangent maps can be computed efficiently. Compared to the second kind coordinates of Owren and Marthinsen [25] the advantage of the present framework is the generality of cases to which the theory can be applied.

References

[1] T. J. Bridges and S. Reich. Computing Lyapunov exponents on a Stiefel manifold. *Physica D*, 156:219–238, 2001. 5.2

[2] M. P. Calvo, A. Iserles, and A. Zanna. Numerical Solution of Isospectral Flows. *Math. of Comp.*, 66:1461–1486, 1997. 5.1

[3] E. Celledoni and A. Iserles. Approximating the exponential from a Lie algebra to a Lie group. *Math. Comp.*, 2000. Posted on March 15, PII S 0025-5718(00)01223-0 (to appear in print).

[4] E. Celledoni and B. Owren. A class of low complexity intrinsic schemes for orthogonal integration. Technical Report Numerics No. 1/2001, The Norwegian University of Science and Technology, Trondheim, Norway, 2001.

[5] E. Celledoni and B. Owren. On the implementation of Lie group methods on the Stiefel manifold. Technical Report Numerics No. 9/2001, The Norwegian University of Science and Technology, Trondheim, Norway, 2001.

[6] M. T. Chu. Inverse eigenvalue problems: Theory and applications, June 2001. A series of lecturs presented at IRMA, CRN, Bari, Italy. 5.1

[7] P. E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *J. Nonlinear. Sci.*, 3:1–33, 1993. 1

[8] P. Davies and N. Higham. A schur-parlett algorithm for computing matrix functions. Technical Report NA Report 404, Manchester Centre for Computational Mathematics, Manchester, England, 2002. 2.3.3, 3.3.1, 3.3.2

[9] 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. 5.2

[10] L. Dieci and E. S. Van Vleck. Computation of a few Lyapunov exponents for continuous and descrete dynamical systems. *Appl. Num. Math.*, 17:275–291, 1995. 5.2, 5.2

[11] U. Dressler. Symmetry properties of the Lyapunov spectra of a class of dissipative dynamical systems with viscous damping. *Phys. Rew. A*, 39(4):2103–2109, 1988. 5.2

[12] K. Engø. On the construction of geometric integrators in the RKMK class. *BIT*, 40(1):41–61, 2000. 1, 2.1

[13] G. H. Golub, C. F. Van, and Loan. *Matrix Computations*. John Hopkins Univ. Press, Baltimore and London, 2 edition, 1989. 2.3.3, 3.3.1, 3.3.1

[14] S. Helgason. *Differential Geometry, Lie Groups and Symmetric Spaces*. Academic Press, 1978. 2.4

[15] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985. 5.1, 5.18

[16] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000. 2.1, 5.2

[17] A. Iserles and S. P. Nørsett. On the solution of linear differential equations in Lie groups. *Phil. Trans. Royal Soc. A*, 357:983–1019, 1999. 1

[18] F. Kang and S. Zai-jiu. Volume-preserving algorithmms for source-free dynamical systems. *Numer. Math.*, 71(4):451–463, 1995. 1

[19] S. Krogstad. A low complexity Lie group method on the Stiefel manifold. *BIT*. To appear. 5.2

[20] D. Lewis and J. C. Simo. Conserving algorithms for the dynamics of Hamiltonian systems of Lie groups. *J. Nonlinear. Sci.*, 4:253–299, 1994. 1, 2.1

[21] H. Munthe-Kaas. Runge–Kutta methods on Lie groups. *BIT*, 38(1):92–111, 1998. 2.1

[22] H. Munthe-Kaas. High order Runge–Kutta methods on manifolds. *Appl. Numer. Math.*, 29:115–127, 1999. 1, 2.1, 2.1, 2.1

[23] H. Munthe-Kaas, G. R. W. Quispel, and A. Zanna. Generalized polar decompositions on Lie groups with involutive automorphisms. *Foundations of Computational Mathematics*, 1(3):297–324, 2001. 1, 2.1, 2.4

[24] B. Owren and A. Marthinsen. Runge–Kutta methods adapted to manifolds and based on rigid frames. *BIT*, 39(1):116–142, 1999. 1

[25] B. Owren and A. Marthinsen. Integration methods based on canonical coordinates of the second kind. *Numer. Math.*, 87(4):763–790, 2001. 1, 2.1, 6

[26] S. T. Smith. Optimization techniques on Riemaniann manifolds. In A. Bloch, editor, *Hamiltonian and Gradient Flows, Algorithms and Control*, volume 3 of *Fields Institute Communications*, pages 113–136. AMS, 1994. 5.1

[27] A. Zanna and H. Z. Munthe-Kaas. Generalized polar decompositions for the approximation of the matrix exponential. *SIAM J. Numer. Anal.*, 23(3):840–862, 2001. 1, 4.1