Numer. Math. (2001) 87: 763–790
Digital Object Identifier (DOI) 10.1007/s002110000211

Numerische
Mathematik

Integration methods based on canonical coordinates of the second kind[*]

Brynjulf Owren, Arne Marthinsen

Department of Mathematical Sciences, Norwegian University of Science and Technology, 7491 Trondheim, Norway; e-mail: {Brynjulf.Owren,Arne.Marthinsen}@math.ntnu.no

Received April 2, 1999 / Revised version received January 18, 2000 /
Published online August 2, 2000 – © Springer-Verlag 2000

Summary. We present a new class of integration methods for differential equations on manifolds, in the framework of Lie group actions. Canonical coordinates of the second kind is used for representing the Lie group locally by means of its corresponding Lie algebra. The coordinate map itself can, in many cases, be computed inexpensively, but the approach also involves the inversion of its differential, a task that can be challenging. To succeed, it is necessary to consider carefully how to choose a basis for the Lie algebra, and the ordering of the basis is important as well. For semisimple Lie algebras, one may take advantage of the root space decomposition to provide a basis with desirable properties. The problem of ordering leads us to introduce the concept of an admissible ordered basis (AOB). The existence of an AOB is established for some of the most important Lie algebras. The computational cost analysis shows that the approach may lead to more efficient solvers for ODEs on manifolds than those based on canonical coordinates of the first kind presented by Munthe-Kaas. Numerical experiments verify the derived properties of the new methods.

Mathematics Subject Classification (1991): 65L05

1 Introduction

The adaptation of Runge–Kutta methods to homogeneous manifolds proposed by Munthe-Kaas in [11] is based on canonical coordinates of the first

[*] This work was in part sponsored by The Norwegian Research Council under contract no. 111038/410, through the SYNODE project. WWW: http://www.math.ntnu.no/num/synode/
Correspondence to: B. Owren

kind. The methods use a Lie algebra action and a Lie group action on a manifold, the two actions being related through the exponential mapping from the Lie algebra to the Lie group. There is also a variety of other methods which are based on this mapping, for instance those by Crouch and Grossman [4,13], and the ones by Zanna [17]. Even if one, in most applications of these methods, assumes that the Lie algebra is of finite dimension, it may still be a challenging computational task to evaluate the exponential map for an arbitrary member of the algebra. If, for instance, the Lie algebra is realized by means of $n \times n$ matrices, one can use general purpose software for computing the matrix exponential, but typically the cost will then be Cn^3 floating point operations where C can be fairly large, say 20–30. A possible remedy for this is to replace $\exp$ by a map that approximates the exponential mapping, but which is cheaper to compute. For certain Lie algebras, one can use the Cayley transform, which under the circumstances above, still costs Cn^3 flops, but where C can be made much smaller. Another problem is that alternative maps may be hard to find. Certain negative results for specific Lie algebras support this observation. A typical one concerns the special linear algebra $\mathfrak{sl}(V)$ of trace-free endomorphisms of a linear space V, and its corresponding Lie group $\mathrm{SL}(V)$ of automorphisms of V with unit determinant. It is proved in [7] that if a function Ψ, analytic at 0, maps $\mathfrak{sl}(V)$ into $\mathrm{SL}(V)$, $\dim(V) > 2$, and satisfies $\Psi(0) = \Psi'(0) = 1$, then $\Psi = \exp$. In a recent paper by Celledoni and Iserles [2] the authors propose to approximate the exponential map by means of splitting techniques. Suppose we need to exponentiate $a \in \mathfrak{g} \subset \mathfrak{gl}(V)$ and let $a = \sum a_i$ with each $a_i \in \mathfrak{g}$. It was observed that if the rank of each a_i (as an endomorphism of V) is small, then its exponential can be computed at low cost. Moreover, their compositions can be computed inexpensively as well.

In this work, we shall consider a fixed basis, say $e_1, \ldots, e_d$, for the Lie algebra, $\mathfrak{g}$, and for the corresponding Lie group G, we shall use canonical coordinates of the second kind. This means that in some neighborhood U of 0 in $\mathfrak{g}$, we define the map

$$(1)\quad \Psi : U \to G, \quad v = \sum_i v_i e_i \mapsto \exp(v_1 e_1) \cdot \exp(v_2 e_2) \cdots \exp(v_d e_d).$$

In some parts of the exposition, it would have cost us little extra effort to replace the function $\exp$ in (1) by an arbitrary smooth map from $U \subseteq \mathfrak{g}$ to G, but to maintain a convenient notation, we shall use the exponential mapping in what follows. We note that the approach with canonical coordinates of the second kind has recently been used by Celledoni and Iserles [3] for approximating the matrix exponential.

The coordinate choice will together with the Lie algebra action on the manifold M, serve to transform the differential system from M to a corresponding system of ODEs in the Lie algebra $\mathfrak{g}$. An implementation of this transformation involves the computation of the inverse of the (trivialized) tangent mapping of Ψ, denoted $d\Psi_u^{-1}$. This amounts to inverting a $d \times d$ linear system in each evaluation of the right hand side of the transformed system, where $d = \dim \mathfrak{g}$. Thus, using general software for this purpose, we must expect a complexity of order d^3. However, comparing with the Munthe-Kaas approach for several of the most interesting Lie algebras, one finds that the extra cost related to the inversion of the tangent mapping should not have arithmetic complexity of higher order than $d^{3/2}$. To overcome this challenge, we have found that the Chevalley basis, known from the structure theory of Lie algebras, can be used. This choice must be combined with a certain ordering of the basis. We will start by considering how the differential equations can be transformed from M to $\mathfrak{g}$, reviewing some results from [11] and [5], in particular we will derive the expression for $d\Psi_u$ involved in this transformation. Next, we consider a criterion on the ordering of the Chevalley basis which allows us to cheaply invert $d\Psi_u$. We can then study, in particular, the case when $\mathfrak{g}$ is a semisimple Lie algebra over $\mathbb{C}$. We also consider solvable Lie algebras, recalling that any Lie algebra can be decomposed into the semidirect product of a semisimple subalgebra of $\mathfrak{g}$ and the radical of $\mathfrak{g}$, by the Levi decomposition [16, p. 224]. Finally, we will present numerical examples which support our claims, and we will make comparisons in terms of flops and accuracy of the derived methods with those of Munthe-Kaas [11].

2 Lie group methods

Suppose that G is a Lie group with Lie algebra $\mathfrak{g}$, and that both are acting on the manifold M through $\Lambda : G \times M \to M$ and $\lambda : \mathfrak{g} \times M \to M$, respectively. Furthermore, suppose that there is a coordinate map $\Psi : \mathfrak{g} \to G$ such that the two actions are related through the equation

$$\lambda(v,p) = \Lambda(\Psi(v),p), \quad v \in \mathfrak{g},\ p \in M.$$

The reader may keep in mind the particular case in which $M = G$, and Λ acts by G on G through left multiplication, i.e. $\Lambda(g,h) = g \cdot h$.

Munthe-Kaas [11] introduces what he calls the generic presentation of ODEs on manifolds in terms of the Lie algebra action λ. His assumption is that the vector field F, which defines the differential equations on M, is related to a map $f : \mathbb{R} \times M \to \mathfrak{g}$ such that

$$y' = F(t,y) = \lambda_*(f(t,y))(y). \tag{2}$$

Here λ_* is a map from $\mathfrak{g}$ to the set of vector fields on M. For $v \in \mathfrak{g}$ it is defined as

$$\lambda_*(v)(p) = \left.\frac{d}{dt}\right|_{t=0} \lambda(tv,p).$$

For our purposes, it is sufficient that λ (and thereby λ_*) is defined locally, but the reader should consult both [11] and [5] for a more detailed discussion of this setting.

The derivative of Ψ at u is a map $\Psi_u' : \mathfrak{g} \to T_{\Psi(u)}G$. We will rather prefer to work with the right trivialized map $d\Psi_u : \mathfrak{g} \to \mathfrak{g}$, related to Ψ_u' through

$$\Psi_u' = {R'}_{\Psi(u)} \circ d\Psi_u.$$

Here R_g is the right translation map $R_g(h) = h \cdot g,\ g, h \in G$, and $R_g' = T_e R_g : \mathfrak{g} \to T_g G$. An elementary modification of a theorem in [11], leads to the following result.

Theorem 1 *Let the circumstances be as described above. For any point $p \in M$, set $\lambda_p(u) = \lambda(u,p)$. Define the vector field $\tilde{f} : \mathbb{R} \times \mathfrak{g} \to \mathfrak{g}$ relative to $p \in M$ as*

$$\tilde{f}(t,u) = d\Psi_u^{-1}(f(t, \lambda(u,p))).$$

Then

$$\lambda_p' \circ \tilde{f} = F \circ \lambda_p,$$

where the composition applies to the second argument of F.

Proof. The proof is almost an exact copy of the corresponding proof in [11] but is included for completeness. We assume here that the Lie algebra $\mathfrak{g}$ is identified with $T_e G$. The existence of a local group action $\Lambda : G \times M \to M$ such that $\Lambda(\Phi(v), p) = \lambda(v,p)$ as presented above is crucial. We write $\Lambda_p(g) = \Lambda(g,p)$, $f_t(u) = f(t,u)$, $\tilde{f}_t(u) = \tilde{f}_t(u)$, and compute

$$\begin{aligned}\lambda_p' \circ \tilde{f}_t(u) &= (\Lambda_p \circ \Phi(u))' \circ \tilde{f}_t(u)\\ &= \Lambda_p' \circ R_{\Phi(u)}' \circ d\Phi_u \circ d\Phi_u^{-1}(f_t \circ \lambda_p(u))\\ &= \Lambda_p' \circ R_{\Phi(u)}' \circ f_t \circ \lambda_p(u).\end{aligned}$$

For any $u, v \in \mathfrak{g}$ we have

$$\begin{aligned}\lambda_*(v)(\lambda_p(u)) &= \left.\frac{d}{dt}\right|_{t=0} \Lambda(\Phi(tv), \Lambda(\Phi(u), p))\\ &= \left.\frac{d}{dt}\right|_{t=0} \Lambda(\Phi(tv) \cdot \Phi(u), p) = \Lambda_p' \circ R_{\Phi(u)}'(v).\end{aligned}$$

So finally we compute

$$\begin{aligned}F \circ \lambda_p(u) &= \lambda_*\big(f_t \circ \lambda_p(u)\big)\big(\lambda_p(u)\big)\\ &= \Lambda_p' \circ R_{\Phi(u)}'\big(f_t \circ \lambda_p(u)\big) = \lambda_p' \circ \tilde{f}_t.\end{aligned}$$

An implication of this theorem is that one can replace the differential system on M by the equation $u' = \tilde{f}(t,u)$ on $\mathfrak{g}$, and the solution of the original equation $y' = F(t,y)$ is obtained simply as $y(t) = \lambda_p(u(t))$. One needs to be concerned with finding efficient methods for computing the action $\lambda_p(u) = \Lambda(\Psi(u),p)$ as well as the inverse tangent map $d\Psi_u^{-1}(v)$ for arbitrary $u, v \in \mathfrak{g}$.

We now suppose that a differential equation on the manifold M is given in the form (2), and that we are solving, according to Theorem 1, the equation

$$u' = d\Psi_u^{-1}(f(t, \lambda(u,p))) \tag{3}$$

by a classical integration method in some neighborhood of $p \in M$. The most popular integration methods can be divided into two classes, the linear multi-step methods and the one-step methods, first and foremost represented by the Runge–Kutta methods. Also, recently the general linear methods have increased their popularity, and offer a third alternative. The fact that the coordinate chart centered at p only yields a local representation of M implies that one generally needs to switch charts throughout the integration, this amounts to altering p in (3). In the case of multi-step methods, this may cause some difficulty, since they carry approximations to the solution and its derivative in several points, and they all have to be transferred to the new coordinate chart whenever a switch is taking place. There are several possibilities in handling this, and we refer to [6] for details. With Runge–Kutta methods the situation is simpler, one can change coordinate chart in each step without problems. All numerical results presented in Sect. 6 will be based on the use of Runge–Kutta methods. The algorithm we obtain is, apart from the coordinate map, the same as the one presented by Munthe-Kaas in [11]. Let $(a_{ij}),\ 1 \le i,j \le s$, be the elements of the Butcher matrix, and let $(b_i),\ 1 \le i \le s$, be the weights. These coefficients can be taken from any classical Runge–Kutta method, no special requirements need to be imposed.

Algorithm 2

```
$y_0 := p$
**for** $i = 1, 2, \ldots, s$
    $u_i := h\sum_{j=1}^{s} a_{i,j}\tilde{k}_j$
    $k_i := f(hc_i, \lambda(u_i, y_0))$
    $\tilde{k}_i := d\Psi_{u_i}^{-1}(k_j)$
**end**
$v := h\sum_{j=1}^{s} b_j\tilde{k}_j$
$y_1 := \lambda(v, y_0)$
*where* $p, y_0, y_1 \in M$, $u_i, k_i, \tilde{k}_i, v \in \mathfrak{g}$.
```

If the coefficients $(a_{i,j})$ come from an explicit Runge–Kutta method, i.e. $a_{i,j} = 0,\ i \leq j,$ then the above algorithm is explicit. In principle, the algorithm may also be used when the coefficients come from an implicit Runge–Kutta method, but the resulting computational costs may then be high.

As opposed to the methods in [11], the cost of computing the correction $\tilde{k}_i = d\Psi_{u_i}^{-1}(k_j)$ does not depend on the order of the Runge–Kutta method, see Tables 1–3 for details regarding the classical Lie algebras. It is difficult to make precise comparisons between the cost of computing the corrections in the above methods and the Munthe-Kaas methods, because the latter make use of Lie brackets as part of the correction, whose computational cost may be hard to quantify. However, an upper bound for cost of computing the commutator between two $n \times n$ matrices is $2n^3$ additions and $2n^3$ multiplications, thus even *one* commutator computed in this way is far more costly than the computation of $d\Psi_u^{-1}$ as proposed for the classical Lie algebras A_ℓ – D_ℓ discussed in Sect. 4.

Still, we believe that the major difference between the cost of the Munthe-Kaas methods and those presented here, lies in computing the coordinate map as a part of the action λ. To perform a comparison, one again needs to make certain assumptions. Suppose that a matrix representation is used for the elements of the Lie group/Lie algebra, and that the coordinate maps are realized as the matrix exponential (composition of matrix exponentials, respectively). Assume furthermore that we use the matrix representations discussed for the Lie algebras A_ℓ – D_ℓ. Then the cost of computing the map (1) is for each case approximately $1 \cdot n^3$ additions and $1 \cdot n^3$ multiplications. In comparison, our experience with the MATLAB function `expm` for computing the corresponding map for canonical coordinates of the first kind is typically a total of $C \cdot n^3$ additions and multiplications. The constant C depends on the size of the matrix elements, but in our experience it usually lies in the range 20–30.

3 Preliminaries

We now assume that Ψ is given by (1), introduce the basis $e_1, \ldots, e_d$ for $\mathfrak{g}$, and compute, for $u = u_1 e_1 + \cdots + u_d e_d$ and $v = v_1 e_1 + \cdots + v_d e_d$,

$$\begin{aligned} d\Psi_u(v) &= \left.\frac{d}{dt}\right|_{t=0} R'_{\Psi(u)^{-1}} \circ \Psi(u + tv) \\ &= \left.\frac{d}{dt}\right|_{t=0} \exp((u_1 + tv_1)e_1) \cdots \exp((u_d + tv_d)e_d)\Psi(u)^{-1} \\ &= v_1 e_1 + \sum_{i=2}^{d} v_i \,\mathrm{Ad}_{\mathrm{e}^{u_1 e_1}} \cdots \mathrm{Ad}_{\mathrm{e}^{u_{i-1} e_{i-1}}}(e_i). \qquad (4) \end{aligned}$$

Here, for any $g \in G$, the operator $\mathrm{Ad}_g : \mathfrak{g} \to \mathfrak{g}$ is defined as

$$\mathrm{Ad}_g = L'_g \circ R'_{g^{-1}},$$

where $L_g : h \mapsto g \cdot h$ and $R_g : h \mapsto h \cdot g$. In (4) and throughout this paper, we will omit the symbol $\circ$ in compositions whenever it is convenient, thus composition of operators will be signified by juxtaposition.

Let $(e_1, \dots, e_d)$ be an ordered basis for a Lie algebra $\mathfrak{g}$, and let $(\varepsilon_1, \dots, \varepsilon_d)$ be the dual basis for $\mathfrak{g}^*$, i.e. $\varepsilon_i(e_j) = \delta_{ij},\ 1 \le i, j \le d$. For any linear operator $A : \mathfrak{g} \to \mathfrak{g}$, we denote by $A^* : \mathfrak{g}^* \to \mathfrak{g}^*$ the transpose, such that for any $\phi \in \mathfrak{g}^*$ and $v \in \mathfrak{g}$,

$$(A^*\phi)(v) = \phi(Av).$$

For each $k = 0, \dots, d$, we define the subspaces

$$V_k = \mathrm{span}\{e_1, \dots, e_k\}, \qquad V_k^c = \mathrm{span}\{e_{k+1}, \dots, e_d\},$$

where we take $V_0 = V_d^c = \{0\}$, and $V_d = V_0^c = \mathfrak{g}$. We also define the projector $P_k : \mathfrak{g} \to V_k^c$ by

$$P_k : \sum_{i=1}^{d} v_i e_i \mapsto \sum_{j=k+1}^{d} v_j e_j,$$

and we let P_0 and P_d equal the identity and zero operator on $\mathfrak{g}$, respectively. For each $u = \sum u_i e_i \in \mathfrak{g}$, we define the linear operators $\widehat{\mathrm{Ad}}_{\mathrm{e}^{u_k e_k}} : \mathfrak{g} \to \mathfrak{g}$ by

$$\widehat{\mathrm{Ad}}_{\mathrm{e}^{u_k e_k}} = I - P_k + \mathrm{Ad}_{\mathrm{e}^{u_k e_k}} P_k, \quad k = 1, \dots, d.$$

Definition 1 *We shall say that the ordered basis* $(e_1, \dots, e_d)$ *is* an admissible ordered basis *(AOB) if, for each* $u = \sum u_j e_j \in \mathfrak{g}$ *and for each* $i = 1, \dots, d-1$, *we have*

$$\mathrm{Ad}_{\mathrm{e}^{u_1 e_1}} \cdots \mathrm{Ad}_{\mathrm{e}^{u_i e_i}} P_i = \widehat{\mathrm{Ad}}_{\mathrm{e}^{u_1 e_1}} \cdots \widehat{\mathrm{Ad}}_{\mathrm{e}^{u_i e_i}} P_i. \tag{5}$$

Since $P_{i-1}(e_i) = e_i,\ i = 1, \dots, d-1$, and $\widehat{\mathrm{Ad}}_{\mathrm{e}^{u_i e_i}}(e_j) = e_j,\ j \le i$, we obtain

Proposition 1 *If the basis* $(e_1, \dots, e_d)$ *is an AOB, it holds that*

$$d\Psi_u = \widehat{\mathrm{Ad}}_{\mathrm{e}^{u_1 e_1}} \cdots \widehat{\mathrm{Ad}}_{\mathrm{e}^{u_{d-1} e_{d-1}}}, \quad \textit{for all } u \in \mathfrak{g}.$$

This result serves as a motivation for the above definition of an AOB. Each of the operators $\widehat{\mathrm{Ad}}_{\mathrm{e}^{u_k e_k}}$, $k = 1, \ldots, d-1$, are invertible whenever $u = \sum u_i e_i$ belongs to a sufficiently small neighborhood of $0 \in \mathfrak{g}$, in which case we have

$$d\Psi_u^{-1} = \widehat{\mathrm{Ad}}^{-1}_{\mathrm{e}^{u_{d-1}e_{d-1}}} \cdots \widehat{\mathrm{Ad}}^{-1}_{\mathrm{e}^{u_1 e_1}}.$$

Suppose that the ordered basis can be chosen such that for some $d_* < d$, $V_{d_*}^c$ is an abelian subalgebra of $\mathfrak{g}$. Then the restriction of $\mathrm{Ad}_{\mathrm{e}^{u_i e_i}}$, $i > d_*$, to $V_{d_*}^c$ is the identity operator on $V_{d_*}^c$ and it follows that $\widehat{\mathrm{Ad}}_{\mathrm{e}^{u_i e_i}}$, $i > d_*$, is the identity operator on all of $\mathfrak{g}$. Thus, for an AOB with this property, we have

$$d\Psi_u^{-1} = \widehat{\mathrm{Ad}}^{-1}_{\mathrm{e}^{u_{d_*}e_{d_*}}} \cdots \widehat{\mathrm{Ad}}^{-1}_{\mathrm{e}^{u_1 e_1}}. \tag{6}$$

Considering Definition 1 it may seem difficult to find an AOB for a Lie algebra $\mathfrak{g}$, or even to verify whether a given ordered basis is an AOB. For this reason we give the following tool for determining an AOB for an arbitrary Lie algebra $\mathfrak{g}$.

Theorem 3 *Let $(e_1, \ldots, e_d)$ be an ordered basis for $\mathfrak{g}$ and suppose that for all pairs of integers (i, m) such that $1 \le m < i \le d-1$ and for all $u = \sum_i u_i e_i \in \mathfrak{g}$, either*

$$P_i^* \mathrm{Ad}^*_{\mathrm{e}^{u_i e_i}} \varepsilon_m = 0,$$

or

$$[e_m, e_n] = 0, \quad m < n < i.$$

Then $(e_1, \ldots, e_d)$ is an AOB for $\mathfrak{g}$.

Proof. We use the shorthand notation $A_j = \mathrm{Ad}_{\mathrm{e}^{u_j e_j}}$ and $\widehat{A}_j = \widehat{\mathrm{Ad}}_{\mathrm{e}^{u_j e_j}}$, and set $Q_j = I - P_j$. Thus, $\widehat{A}_j = A_j + (I - A_j)Q_j$, and we compute, for any $w \in \mathfrak{g}$,

$$\begin{aligned}\widehat{A}_1 \cdots \widehat{A}_i P_i w = A_1 \cdots A_i P_i w \\ + \sum_{j=1}^{\mathrm{i}-1} \widehat{A}_1 \cdots \widehat{A}_{j-1}(I - A_j) Q_j A_{j+1} \cdots A_i P_i w.\end{aligned}$$

Set

$$Z_r := \sum_{j=1}^{r-1} \widehat{A}_1 \cdots \widehat{A}_{j-1}(I - A_j) Q_j A_{j+1} \cdots A_r P_r, \quad 1 \le r \le i-1.$$

We wish to prove, by induction on i, that Z_i vanishes on all of $\mathfrak{g}$, for each $1 \le i \le d-1$, under the assumptions of the theorem. This holds trivially for $i=1$. Suppose that for some $i>1$, $Z_r = 0,\ 1 \le r \le i-1$. Then set

$$A_i P_i w = \sum_{m=1}^{d} \varepsilon_m (A_i P_i w) e_m,$$

so that

$$Z_i w = \sum_{m=1}^{d} \varepsilon_m (A_i P_i w) \sum_{j=1}^{\mathrm{i}-1} \widehat{A}_1 \cdots \widehat{A}_{j-1} (I - A_j) Q_j A_{j+1} \cdots A_{i-1} e_m.$$

We now split the outer sum into two parts $\sum_{m=1}^{\mathrm{i}-1} + \sum_{m=i}^{d}$. In the latter sum, we may replace e_m by $P_{i-1}e_m$ and discard the last term of the inner sum, $j = i-1$, since $Q_{i-1}e_m = 0$, $m \ge i$. Thus we can invoke the induction hypothesis, $Z_{i-1} = 0$, and conclude that

$$Z_i w = \sum_{m=1}^{\mathrm{i}-1} \varepsilon_m (A_i P_i w) \sum_{j=1}^{\mathrm{i}-1} \widehat{A}_1 \cdots \widehat{A}_{j-1} (I - A_j) Q_j A_{j+1} \cdots A_{i-1} e_m.$$

Let $1 \le m' \le i-1$ be any integer such that $\varepsilon_{m'}(A_i P_i w) \ne 0$ for some $w \in \mathfrak{g}$, or equivalently, $P_i^* A_i^* \varepsilon_{m'} \ne 0$. The assumption $[e_{m'}, e_n] = 0,\ m' \le n \le i-1$, implies that $A_n e_{m'} = e_{m'}$, so if we split the inner sum in two pieces for $m = m'$, we get

$$\sum_{j=1}^{m'-1} \widehat{A}_1 \cdots \widehat{A}_{j-1} (I - A_j) Q_j A_{j+1} \cdots A_{m'-1} e_{m'} \\ + \sum_{j=m'}^{\mathrm{i}-1} \widehat{A}_1 \cdots \widehat{A}_{j-1} (I - A_j) Q_j e_{m'}.$$

The first sum vanishes by the induction hypothesis, $Z_{m'-1} = 0$, since $e_{m'} = P_{m'-1} e_{m'}$. In the second sum we note that since $j \ge m'$, $Q_j e_{m'} = e_{m'}$, and $(I - A_j) e_{m'} = 0$ since $[e_j, e_{m'}] = 0$. This concludes the proof.

When the basis can be chosen such that ad_{e_i} is nilpotent for many of the basis elements e_i, it is useful to recall the relation $\mathrm{Ad}_{\mathrm{e}^{u_j e_j}} = \exp(\mathrm{ad}_{u_j e_j})$, and we can rephrase Theorem 3 as follows

Corollary 1 *Let $(e_1, \ldots, e_d)$ be an ordered basis for $\mathfrak{g}$ and suppose that for all $1 \le m < i \le d-1$, $k \in \mathbb{N}$, either*

$$P_i^* \left(\mathrm{ad}_{e_i}^{k} \right)^* \varepsilon_m = 0,$$

or

$$[e_m, e_n] = 0, \quad m < n < i.$$

Then $(e_1, \dots, e_d)$ *is an AOB for* $\mathfrak{g}$.

4 The semisimple case

In this section we consider semisimple Lie algebras over $\mathbb{C}$. Most of the tools we use are valid for any algebraically closed field. We will use, without proofs, several well known results from the structure theory of semisimple Lie algebras, the reader may consult the texts [10,16,8] for details and proofs, and also [1] which gives a good introduction without too many details.

We start by recalling that for any Lie algebra, the *Killing form* is defined as the bilinear form $\kappa(u,v) = \mathrm{Tr}(\mathrm{ad}_u \circ \mathrm{ad}_v)$, for $u, v \in \mathfrak{g}$. A Lie algebra is semisimple if and only if κ is nondegenerate. Every semisimple Lie algebra contains semisimple elements ($x \in \mathfrak{g}$ is semisimple if ad_x is diagonalizable). A subalgebra $\mathfrak{h}$ is *toral* if all its elements are semisimple. Moreover, if $\mathfrak{h}$ is not properly contained in any other toral subalgebra, it is called a *maximal toral subalgebra* (MTS). For semisimple Lie algebras, an MTS always exists. A toral subalgebra of $\mathfrak{g}$ is abelian, and the eigenspace of each $\mathrm{ad}_h,\ h \in \mathfrak{h}$, equals all of $\mathfrak{g}$. Thus, the operators $\mathrm{ad}_h,\ h \in \mathfrak{h}$, form a commuting family of linear transformations of $\mathfrak{g}$ and are therefore *simultaneously diagonalizable*. In other words, there exists a full set of eigenvectors shared by all $\mathrm{ad}_h,\ h \in \mathfrak{h}$. For each such eigenvector $x \in \mathfrak{g}$, there is an element α in the dual space $\mathfrak{h}^*$ of $\mathfrak{h}$ such that $[h, x] = \alpha(h)x$ for each $h \in \mathfrak{h}$. In particular, the 0-functional corresponds to $\{x \in \mathfrak{g} : [h,x] = 0 \ \forall\ h \in \mathfrak{h}\}$, the centralizer of $\mathfrak{h}$ in $\mathfrak{g}$. It can be proved that $\mathfrak{h}$ equals its centralizer in $\mathfrak{g}$. The $\alpha \neq 0$ in $\mathfrak{h}^*$ defined as above are called *roots*, we denote by Φ the set of roots. For each $\alpha \in \mathfrak{h}^*$, we let $\mathfrak{g}_\alpha = \{x \in \mathfrak{g} : [h,x] = \alpha(h)x,\ \forall\ h \in \mathfrak{h}\}$, and we can thus write down the *root space decomposition* or *Cartan decomposition* of $\mathfrak{g}$ as a direct sum

$$\mathfrak{g} = \mathfrak{h} \oplus \coprod_{\alpha \in \Phi} \mathfrak{g}_\alpha.$$

The reader should note that the elements of Φ are not generally linearly independent in $\mathfrak{h}^*$. For future use we define $\overline{\Phi} = \Phi \cup \{0\}$.

We summarize some properties of the roots, the proofs can be found in [10] and [8].

Theorem 4

a. Each $\mathfrak{g}_\alpha$, $\alpha \in \Phi$, is one-dimensional.
b. If $x \in \mathfrak{g}_\alpha$ then ad_x is nilpotent.
c. For arbitrary $\alpha, \beta \in \mathfrak{h}^*$, we have $[\mathfrak{g}_\alpha, \mathfrak{g}_\beta] \subseteq \mathfrak{g}_{\alpha+\beta}$. Moreover, if $\alpha,\ \beta,\ \alpha + \beta \in \Phi$, then $[\mathfrak{g}_\alpha, \mathfrak{g}_\beta] = \mathfrak{g}_{\alpha+\beta}$.
d. $\alpha \in \Phi \ \Rightarrow \ -\alpha \in \Phi$.
e. The restriction of the Killing form to $\mathfrak{h}$ is nondegenerate.

A choice of basis for $\mathfrak{g}$ known as the *Chevalley* basis is obtained by choosing one basis vector for each subspace $\mathfrak{g}_\alpha$ along with a particular basis for $\mathfrak{h}$.

The existence of a *base* is significant for our further use of roots. A subset Δ of Φ is a base if

1. $\Delta = (\alpha_1, \dots, \alpha_\ell)$ is a basis of $\mathfrak{h}^*$.
2. Each root $\alpha \in \Phi$ can be written as

$$\alpha = \sum_{i=1}^{\ell} r_i \alpha_i$$

with integer coefficients r_i that are all nonnegative or nonpositive.

A base exists for any system of roots. The above properties of a base, together with **d.** of Theorem 4, allow us to split Φ into two disjoint subsets, Φ^+ and Φ^- of the same cardinality, $\Phi = \Phi^+ \cup \Phi^-$. The positive roots Φ^+ are those which can be written as a nonnegative linear combination of the elements of Δ.

The fact that the Killing form is nondegenerate on $\mathfrak{h}$ gives us a way to identify $\mathfrak{h}$ and $\mathfrak{h}^*$ in a unique way. We may associate $\phi \in \mathfrak{h}^*$ with $t_\phi \in \mathfrak{h}$ if $\phi(h) = \kappa(t_\phi, h)$ for all $h \in \mathfrak{h}$. In this way, we also obtain a bilinear form on $\mathfrak{h}^*$, defined for any $\alpha, \beta \in \mathfrak{h}^*$ as $(\alpha, \beta) = \kappa(t_\alpha, t_\beta)$. Given a base $\Delta = (\alpha_1, \dots, \alpha_\ell)$ this identification determines a basis for $\mathfrak{h}$, we take it to be the set $(h_{\alpha_1}, \dots, h_{\alpha_\ell})$

$$h_{\alpha_i} = \frac{2t_{\alpha_i}}{\kappa(t_{\alpha_i}, t_{\alpha_i})}. \tag{7}$$

We may now express any basis for $\mathfrak{h}$ to be used in the AOB as a linear combination of the h_{α_i}. The choice which is optimal with respect to computational cost may depend on the Lie algebra, we may in each case use Proposition 2 below to search for an optimal basis for $\mathfrak{h}$. First, with a given base for a semisimple Lie algebra, the numbers

$$\mathbf{C}_{i,j} = \langle \alpha_i, \alpha_j \rangle = \frac{2(\alpha_i, \alpha_j)}{(\alpha_j, \alpha_j)}, \quad 1 \le i, j \le \ell,$$

are called the Cartan numbers. It can be shown that they are all integers, and $\mathbf{C} = (\mathbf{C}_{i,j})_{i,j=1}^{\ell} \in \mathbb{Z}^{\ell \times \ell}$ is called the Cartan matrix. It turns out that, up to isomorphisms, $\mathbf{C}$ determines the Lie algebra completely. The structure of Lie algebras in terms of their root systems is of course studied extensively in the literature to which we refer for more details. However, we just remark here that the number of possible root systems are limited, and in the sequel we shall study the most important of them. Before we proceed to this point, we shall prove some general results which will be useful later.

Proposition 2 *Let $\beta = \sum_{i=1}^{\ell} r_i \alpha_i \in \Phi$ and let $0 \neq e_\beta \in \mathfrak{g}_\beta$. Then, for any basis element h_{α_j} under the above identification, we have*

$$\operatorname{ad}_{e_\beta}(h_{\alpha_j}) = -\sum_{i=1}^{\ell} r_i \mathbf{C}_{i,j} e_\beta.$$

Proof. We compute

$$\begin{aligned}
\operatorname{ad}_{e_\beta}(h_{\alpha_j}) &= -\beta(h_{\alpha_j})e_\beta = -\sum_{i=1}^{\ell} r_i \alpha_i(h_{\alpha_j}) e_\beta \\
&= -\sum_{i=1}^{\ell} r_i \frac{2}{\kappa(t_{\alpha_j}, t_{\alpha_j})} \alpha_i(t_{\alpha_j}) e_\beta \\
&= -\sum_{i=1}^{\ell} r_i \frac{2(\alpha_i, \alpha_j)}{(\alpha_j, \alpha_j)} e_\beta \\
&= -\sum_{i=1}^{\ell} r_i \mathbf{C}_{i,j} e_\beta.
\end{aligned}$$

We are now in a position to characterize an AOB in terms of the root system.

Theorem 5 *Let $\{\beta_1, \ldots, \beta_{d_*}\}$, $d_* = d - \ell$, be the set of roots Φ for a semisimple Lie algebra $\mathfrak{g}$. Suppose that a Chevalley basis is ordered as*

$$(e_{\beta_1}, \ldots, e_{\beta_{d_*}}, h_1, \ldots, h_\ell)$$

where $e_{\beta_i} \in \mathfrak{g}_{\beta_i}$, and $(h_1, \ldots, h_\ell)$ is a basis for $\mathfrak{h}$. Such an ordered basis is an AOB if

$$\begin{aligned}
&k\beta_i + \beta_s = \beta_m,\ m < i < s \leq d_*, k \in \mathbb{N} \\
&\qquad\qquad \Rightarrow \quad \beta_m + \beta_n \notin \overline{\Phi},\ m < n \leq i - 1.
\end{aligned} \tag{8}$$

Proof. We may check that the conditions in Corollary 1 are satisfied. We use the convention $e_i := e_{\beta_i}$, $1 \leq i \leq d_*$, and $e_{d_*+j} := h_j$, $1 \leq j \leq \ell$, whenever it is convenient. For $e_i \in \mathfrak{h}$, $\operatorname{ad}_{e_i}^k P_i = 0$ for all $k \in \mathbb{N}$, so the first condition of the corollary applies for all $m < i$. Next, suppose that $e_i \in \mathfrak{g}_{\beta_i}$ for some root β_i. We compute for arbitrary $w \in \mathfrak{g}$, $k \in \mathbb{N}$, and $m < i$

$$\varepsilon_m(\mathrm{ad}_{e_i}^k P_i w) = \sum_{s=i+1}^{d} \varepsilon_s(w)\varepsilon_m(\mathrm{ad}_{e_i}^k e_s) = \sum_{s=i+1}^{d_*} \varepsilon_s(w)\varepsilon_m(\mathrm{ad}_{e_i}^k e_s).$$

The last equality results from the fact that whenever $s > d_*$, we have $e_s \in \mathfrak{h}$, and thus $[e_i, e_s] \in \mathfrak{g}_{\beta_i}$ and $\mathrm{ad}_{e_i}^k e_s = 0$ for $k > 1$. From property **c.** in Theorem 4 we conclude that the only cases in which $v := \mathrm{ad}_{e_i}^k e_s \neq 0$ are when $\gamma := k\beta_i + \beta_s \in \overline{\Phi}$. If $\gamma = 0$, then $v \in \mathfrak{h}$, thus $\varepsilon_m(v) = 0$. In fact, we obtain a nonzero term in the above sum if and only if there is an s, with $i < s < d_*$ such that $\beta_m = k\beta_i + \beta_s \in \Phi$. In other words, this condition corresponds precisely to $P_i^* \left(\mathrm{ad}_{e_i}^k\right)^* \varepsilon_m \neq 0$. In such a case, we need to impose the condition that $[e_m, e_n] = 0,\ m < n < i$, and this means that $\beta_m + \beta_n \notin \overline{\Phi},\ m < n < i$.

Note that since all $e_1, \ldots, e_{d_*}$ are ad-nilpotent, it suffices to check that (8) holds for integers $1 \leq k \leq k_*$, where k_* is the smallest integer such that $\mathrm{ad}_{e_i}^{k_*+1} = 0$.

Proposition 3 *If an AOB $(e_1, \ldots, e_d)$ for $\mathfrak{g}$ is ordered as above, and if $e_i \in \mathfrak{g}_{\beta_i}$, $\beta_i \in \Phi$, then*

$$\widehat{\mathrm{Ad}}_{\mathrm{e}^{u_i e_i}}^{-1} = I + \sum_{k=1}^{k_*} \frac{(-u_i)^k}{k!} \mathrm{ad}_{e_i}^k P_i,$$

where $\mathrm{ad}_{e_i}^m = 0$ for $m > k_*$.

Proof. Note that with the ordering above, $\mathrm{ad}_{e_i}^k P_i\, \mathrm{ad}_{e_i}^r P_i = \mathrm{ad}_{e_i}^{k+r} P_i$. Now

$$\widehat{\mathrm{Ad}}_{\mathrm{e}^{u_i e_i}} = (I - P_i) + \exp(\mathrm{ad}_{u_i e_i})P_i = I + \sum_{k=1}^{k_*} \frac{u_i^k}{k!} \mathrm{ad}_{e_i}^k P_i.$$

By direct computation we get the required result.

In the discussion that follows, we shall always assume that $\Delta = (\alpha_1, \ldots, \alpha_\ell)$ is a base, and we define the elements of $\beta_{i,j}$ of $\mathfrak{h}^*$ as

$$\beta_{i,j} = \sum_{k=i}^{j} \alpha_k, \quad 1 \leq i \leq j \leq \ell.$$

4.1 The Lie algebra A_ℓ

This Lie algebra is commonly represented as the set of trace free complex $(\ell+1)\times(\ell+1)$ matrices and is denoted $\mathfrak{sl}(\ell+1,\mathbb{C})$. The MTS is the ℓ-dimensional subalgebra of diagonal matrices. The set of positive roots is [16, p. 296]

$$\{\beta_{i,j}, 1 \le i \le j \le \ell\}.$$

It is obtained from the Cartan matrix

$$\mathbf{C}_{A_\ell} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}.$$

Letting $\mathbf{e}_r$ be the rth canonical unit vector in $\mathbb{C}^{\ell+1}$, we find that $\mathfrak{g}_{\beta_{i,j}}$ is spanned by $\mathbf{e}_i\mathbf{e}_{j+1}^{\mathrm{T}}$, $1 \le i \le j \le \ell$. For the negative roots $-\beta_{i,j}$, we have $\mathbf{e}_{j+1}\mathbf{e}_i^{\mathrm{T}} \in \mathfrak{g}_{-\beta_{i,j}}$. Finally, as a basis for $\mathfrak{h}$, we may for instance take the elements $\mathbf{e}_i\mathbf{e}_i^{\mathrm{T}} - \mathbf{e}_{i+1}\mathbf{e}_{i+1}^{\mathrm{T}}$, $1 \le i \le \ell$. In particular, we immediately obtain a real realization $\mathfrak{sl}(\ell+1,\mathbb{R})$ where all the roots belong. In consequence, everything we do here with $\mathfrak{sl}(\ell+1,\mathbb{C})$ also holds for the corresponding representation of $\mathfrak{sl}(\ell+1,\mathbb{R})$.

We consider how the basis can be ordered such that an AOB results. With the convention $e_\beta \in \mathfrak{g}_\beta$, $\beta \in \varPhi$, $\mathfrak{h} = \mathrm{span}(e_{\mathfrak{h}_1}, \dots, e_{\mathfrak{h}_\ell})$, let

$$B = (e_{\beta_{i_1,j_1}}, \dots, e_{\beta_{i_m,j_m}}, e_{-\beta_{i_1,j_1}}, \dots, e_{-\beta_{i_m,j_m}}, e_{\mathfrak{h}_1}, \dots, e_{\mathfrak{h}_\ell}),$$

where $i_1 \le i_2 \le \cdots \le i_m$ and $m = \ell(\ell-1)/2$. We claim that B is an AOB for $\mathfrak{sl}(\ell+1,\mathbb{C})$, and prove it by using Theorem 5. Let $\varPhi$ be ordered in correspondence with B and consider first a positive root $\beta_{i_r,j_r} = \alpha_{i_r} + \cdots + \alpha_{j_r}$. If $\alpha \in \varPhi$ is such that $\gamma = k\beta_{i_r,j_r} + \alpha \in \varPhi$ with $\beta_{i_r,j_r} < \alpha$, $k \in \mathbb{N}$, then $\gamma = \beta_{i_s,j_s}$ with $i_s \ge i_r$. Thus, either $\gamma > \beta_{i_r,j_r}$ or $i_s = i_r$, but in the latter case, $\gamma + \beta_{i_r,j} \notin \overline{\varPhi}$ for all $j > i_r$. Similarly, for negative roots, $-\beta_{i_r,j_r} = -\alpha_{i_r} - \cdots - \alpha_{j_r}$. Again, if $\gamma = -\beta_{i_r,j_r} + \alpha \in \varPhi$ with $\alpha > \beta_{i_r,j_r}$ then $\gamma = -\beta_{i_s,j_s}$ with $i_s \ge i_r$ and we conclude that B above is an AOB for $\mathfrak{g}$.

In computing $d\varPsi_u^{-1}(v)$, we may apply (6) together with Proposition 3 with $k_* = 2$. In fact, since the coefficient of α_k in the expansion of any root is either $0, 1$ or -1, it is clear that $k\alpha + \beta$, $\alpha \in \varPhi$, $\beta \in \overline{\varPhi}$, is neither 0 nor contained in $\varPhi$ if $k > 2$. In particular, we notice that $\mathrm{rank}(\mathrm{ad}_{e_i}^2) = 1$ whenever e_i corresponds to a root β ($e_i \in \mathfrak{g}_\beta$), since the only root α satisfying $2\beta + \alpha \in \overline{\varPhi}$ is $\alpha = -\beta$. We next consider the computational complexity involved in computing $d\Psi_u^{-1}(v)$ for $\mathfrak{sl}(\ell+1,\mathbb{C})$. We split the computation into smaller tasks, and begin by studying the cost involved in computing, for arbitrary $w \in \mathfrak{g}$, the expression

$$(9) \quad \widehat{\mathrm{Ad}}^{-1}_{\mathrm{e}^{u_k e_k}}(w) = w - u_k\,[e_k, w] + \frac{1}{2}u_k^2\,[e_k,[e_k,w]], \quad 1 \le i \le d_*,$$

where we assume that w is given in the form $w = \sum_{j=1}^{d} w_j e_j$. Suppose that $e_k := e_\beta \in \mathfrak{g}_\beta$, where $\beta \in \Phi$ is of the form $\beta = \pm\beta_{i,j}$. We identify the basis vectors that do not commute with e_β, since only those will affect w. We begin by choosing as a basis for $\mathfrak{h}$, the vectors $h_i = h_{\alpha_i},\ i = 1,\dots,n$, defined by (7). By Proposition 2 and the Cartan numbers, we see that $[e_\beta, h_n] = 0$ unless $n = j, j-1, i, i-1$ (otherwise the coefficient multiplying e_β is 0). Thus, we obtain at most 4 updates caused by the elements of $\mathfrak{h}$, the cost is one addition and one multiplication for each. Next consider $\mathrm{ad}^{\,k}_{e_\beta}(e_{-\beta})$, $k = 1, 2$. Note that, due to the ordering of the basis, this case only occurs when $\beta \in \Phi^+$. From a result in [10, p. 37] we conclude that $[e_\beta, e_{-\beta}] = h_\beta \in \mathfrak{h}$, $h_\beta = 2t_\beta/\kappa(t_\beta,t_\beta)$ under the identification of $\mathfrak{h}$ and $\mathfrak{h}^*$ introduced above. Hence, since $\beta = \beta_{i,j} = \sum_{n=i}^{j} \alpha_n$, we deduce that $h_\beta = \sum_{n=i}^{j} h_{\alpha_n}$. So $j - i + 1$ updates result, requiring each one addition and one multiplication. When $k = 2$, we compute

$$[e_\beta, h_\beta] = -\beta(h_\beta)\,e_\beta = -\langle\beta,\beta\rangle\,e_\beta = -2\,e_\beta.$$

This -2 cancels the factor $1/2$ in (9), thus in this update, we need two multiplications and one addition. Finally, consider $[e_\beta, e_\alpha]$, where $\alpha \ne -\beta$ is an arbitrary root. If $\beta = \beta_{i,j} \in \Phi^+$, then α is of the form

$$\begin{array}{ll} \beta_{j+1,m},\ j+1 \le m \le \ell, & \ell - j \text{ elements},\\ -\beta_{i,m},\ \ i \le m \le \ell,\ m \ne j, & \ell - i \text{ elements},\\ -\beta_{m,j},\ \ 1 \le m \le j,\ m \ne i, & j - 1 \text{ elements}. \end{array}$$

If $\beta = -\beta_{i,j} \in \Phi^-$, then $\alpha \in \Phi^-$ is of the form $\alpha = -\beta_{j+1,m},\ j+1 \le m \le \ell$, thus there are $\ell - j$ noncommuting elements. Each of these updates costs one addition and one multiplication.

For this particular Lie algebra, the above analysis will lead to the precise computational cost of computing $d\Psi_u^{-1}$ in terms of arithmetic operations. However, for other Lie algebras this counting process is more complicated, especially since there may or may not be a constant multiplying the u_i (u_i^2 resp.) in each update, thereby in some cases adding to the number of multiplications. We therefore choose to give the number of updates required for each evaluation of $d\Psi_u^{-1}$, and we divide the updates into categories as

Table 1. Number of updates needed in the computation of $d\Psi_u^{-1}(v)$ for A_ℓ

Elements of $\mathfrak{h}$	$4\ell^2$
Reflected root	$\frac{1}{6}\ell^3 + \frac{1}{2}\ell^2 + \frac{1}{3}\ell$
$\mathrm{ad}\,^2_{u_\beta e_\beta}$	$\frac{1}{2}\ell^2 + \frac{1}{2}\ell$
All other roots	$\ell^3 - \ell$
Total	$\frac{7}{6}\ell^3 + 5\ell^2 - \frac{1}{6}\ell$

described in Table 1. The result in each category is obtained by summing up the updates for all operators $\widehat{\mathrm{Ad}}\,^{-1}_{\mathrm{e}^{u_j e_j}}$ that occur in the expression for $d\Psi_u^{-1}$.

Note in particular, that the ℓ^3 terms come from the categories "All other roots" and "Reflected root", where the cost is one addition and one multiplication for each update.

4.2 The Lie algebra C_ℓ, $l \geq 3$

The symplectic Lie algebra $\mathfrak{sp}(2\ell, \mathbb{C})$ has the root system of C_ℓ and a usual representation is

$$\mathfrak{sp}(2\ell, \mathbb{C}) = \{u \in \mathfrak{gl}(n, \mathbb{C}) : u^{\mathrm{T}} J + Ju = 0\}, \quad J = \begin{bmatrix} 0 & I_\ell \\ -I_\ell & 0 \end{bmatrix}.$$

One easily checks that such matrices u must be of the form

$$u = \begin{bmatrix} Q & M \\ N & -Q^{\mathrm{T}} \end{bmatrix}, \quad M = M^{\mathrm{T}},\ N = N^{\mathrm{T}}.$$

We let, as before, $\alpha_1, \ldots, \alpha_\ell$ be a base. The positive roots in C_ℓ are obtained from the Cartan matrix

$$\mathbf{C}_{C_\ell} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -2 & 2 \end{bmatrix},$$

they can be characterized in terms of $\beta_{i,j} = \sum_{n=i}^{j} \alpha_n$ as follows [16, p. 301]

$$\Phi^+ = \{\beta_{i,j},\ 1 \leq i \leq j \leq \ell\} \bigcup \{\beta_{i,\ell} + \beta_{j,\ell-1},\ 1 \leq i \leq j \leq \ell - 1\}.$$

Thus, the dimension of $\mathfrak{sp}(2\ell, \mathbb{C})$ equals $2\ell^2 + \ell$. Note in particular that all roots are of the form $\alpha = \sum_{n=i_\alpha}^{j_\alpha} r_{\alpha,n} \alpha_n$ with each $r_{\alpha,n} \neq 0$ having the same sign. We use this notation to introduce an ordering of $\varPhi$ according to the following rules.

1. $\alpha_+ \in \varPhi^+,\ \alpha_- \in \varPhi^- \ \Rightarrow\ \alpha_+ < \alpha_-$
2. If $\alpha = \sum_{n=i_\alpha}^{j_\alpha} r_{\alpha,n}\alpha_n,\ \beta = \sum_{n=i_\beta}^{j_\beta} r_{\beta,n}\alpha_n,\ \alpha,\beta \in \varPhi^+\ (\varPhi^- \text{ resp.})$ then $\alpha < \beta \ \Leftarrow\ j_\alpha > j_\beta.$

There always exist orderings satisfying 1 and 2, but generally more than one. We claim that the corresponding ordering of the basis leads to an AOB. Note first that if a coefficient $r_{\alpha,n}$ is such that $|r_{\alpha,n}| > 1$ then $n < \ell$ and $|r_{\alpha,\ell}| = 1$. We suppose that $\alpha \in \varPhi^+$ and use Theorem 5. If $\gamma = k\alpha + \beta$ with $\beta > \alpha$, then either $\gamma \in \varPhi^-$, thus $\gamma > \alpha$ or else, $j_\gamma \leq j_\alpha$. In the latter case, either $\gamma > \alpha$, or $j_\gamma = j_\alpha$, but then all $\delta \in \varPhi$ satisfying $\gamma < \delta < \alpha$ are such that $\gamma + \delta \notin \overline{\varPhi}$. The argument for $\alpha \in \varPhi^-$ is similar, and we conclude that the ordering above leads to an AOB.

As a choice of basis for $\mathfrak{h}$ in the case, we recommend to use the vectors $(h_1, \ldots, h_\ell)$, obtained as linear combinations of the h_{α_j} defined in (7), as follows:

$$h_i = \sum_{j=i}^{\ell} h_{\alpha_j} - \frac{1}{2}h_{\alpha_\ell}, \quad i = 1, \ldots, \ell.$$

This particular choice causes, for all $\beta \in \varPhi$, $[e_\beta, h_i] = 0$ for a minimum of $\ell - 2$ members of $\{h_1, \ldots, h_\ell\}$, according to Proposition 2.

It is easy to check that if $\alpha \in \varPhi$, then the corresponding basis element e_α satisfies $\mathrm{ad}_{e_\alpha}^{k} = 0$ for $k > 2$. Thus, we can use Proposition 3 with $k_* = 2$ for computing $\widehat{\mathrm{Ad}}^{-1}_{e^{u_i e_i}}$. Moreover, one can prove that $\mathrm{rank}(\mathrm{ad}_{e_\alpha}^{2})$ is either 1 or 3 for each $\alpha \in \varPhi$. Certainly, $0 \neq \mathrm{ad}_{e_\alpha}^{2}(e_{-\alpha}) \in \mathrm{span}(e_\alpha)$ for any $\alpha \in \varPhi$. We proceed by making the change of variables $\alpha_i = \lambda_i - \lambda_{i+1},\ 1 \leq i \leq \ell - 1$, $\alpha_\ell = 2\lambda_\ell$. Then, the roots have the form $\pm(\lambda_i - \lambda_j),\ 1 \leq i < j \leq \ell$, and $\pm(\lambda_p + \lambda_q),\ 1 \leq p \leq q \leq \ell$. Thus, the remaining possibilities for obtaining $2\alpha + \beta \in \overline{\varPhi}$ are: If $\alpha = \pm(\lambda_i - \lambda_j),\ 1 \leq i < j \leq \ell$, then $\beta = \pm 2\lambda_j$ or $\beta = \mp 2\lambda_i$. If $\alpha = \pm(\lambda_p + \lambda_q),\ 1 \leq p < q \leq \ell$, then $\beta = \mp 2\lambda_p$ or $\mp 2\lambda_q$. When $\alpha = 2\lambda_p,\ 1 \leq p \leq \ell$, we get $\mathrm{rank}(\mathrm{ad}_{e_\alpha}^{2}) = 1$. We shall omit the details about the cost of computing $d\varPsi_u^{-1}(v)$. The situation is similar, although somewhat more complicated than for $\mathfrak{sl}(\ell+1, \mathbb{C})$. It is for instance no longer true that whenever $\gamma = \alpha + \beta \in \varPhi$, $[e_\alpha, e_\beta] = \pm e_\gamma$, in other words, one sometimes needs to multiply the result with an appropriate factor. This complicates the counting of arithmetic operations. We will give more precise results about the computational cost of computing $d\varPsi_u^{-1}(v)$ in the numerical

Table 2. Number of updates needed in the computation of $d\varPsi_u^{-1}(v)$ for C_ℓ

Elements of $\mathfrak{h}$	$4\ell^2 - 2\ell$
Reflected root	$2\ell^2 - \ell$
$\mathrm{ad}^2_{u_\beta e_\beta}$	$\frac{5}{2}\ell^2 - \frac{3}{2}\ell$
All other roots	$4\ell^3 - 6\ell^2 + 2\ell$
Total	$4\ell^3 + \frac{5}{2}\ell^2 - \frac{5}{2}\ell$

experiment section. The number of updates is given in Table 2. Depending on the type of root, an update costs one addition and 1–3 multiplications.

Note, in particular, that the dominant term, $4\ell^3$, comes from the category “All other roots”, where the cost is one addition and one multiplication for each update.

4.3 *The Lie algebra B_ℓ, $\ell \geq 1$*

The orthogonal Lie algebra $\mathfrak{so}(2\ell+1,\mathbb{C})$, has the root system of B_ℓ and can be represented as matrices in $\mathbb{C}^{(2\ell+1)\times(2\ell+1)}$ of the form

$$\begin{bmatrix} 0 & a^{\mathrm{T}} & b^{\mathrm{T}} \\ -b & Q & M \\ -a & N & -Q^{\mathrm{T}} \end{bmatrix}, \; a, b \in \mathbb{C}^\ell, \; M, N, Q \in \mathbb{C}^{\ell\times\ell}, \; M^{\mathrm{T}} = -M, \; N^{\mathrm{T}} = -N.$$

It is also possible to represent elements of $\mathfrak{so}(2\ell+1,\mathbb{C})$ as complex skew-symmetric $(2\ell+1)\times(2\ell+1)$ matrices, but it leads to a more complicated form of the basis vectors e_α corresponding to roots.

The Cartan matrix is

$$\mathbf{C}_{B_\ell} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -2 \\ & & & -1 & 2 \end{bmatrix},$$

which yields the positive roots [16, p. 304]

$$\varPhi^+ = \{\beta_{i,j},\ 1 \leq i \leq j \leq \ell\} \bigcup \{\beta_{i,\ell} + \beta_{j,\ell},\ 1 \leq i < j \leq \ell\}.$$

We therefore see that the dimension of $\mathfrak{so}(2\ell+1,\mathbb{C})$ is $2\ell^2+\ell$. Also in this case, any root α can be written in the form $\alpha = \sum_{n=i_\alpha}^{j_\alpha} r_{\alpha,n}\alpha_n$ with each

$r_{\alpha,n} \neq 0$ having the same sign. With this notation we propose the following ordering of Φ.

1. $\alpha_+ \in \Phi^+,\ \alpha_- \in \Phi^- \ \Rightarrow\ \alpha_+ < \alpha_-$
2. If $\alpha = \sum_{n=i_\alpha}^{j_\alpha} r_{\alpha,n}\alpha_n,\ \beta = \sum_{n=i_\beta}^{j_\beta} r_{\beta,n}\alpha_n,\ \alpha,\beta \in \Phi^+$ (Φ^- resp.) then $\alpha < \beta \ \Leftarrow\ j_\alpha < j_\beta.$

One can prove that such an ordering corresponds to an AOB, the argument is almost identical to that for $\mathfrak{sp}(2\ell,\mathbb{C})$.

As basis $(h_1,\ldots,h_\ell)$ for $\mathfrak{h}$, we recommend to use

$$h_i = \sum_{j=i}^{\ell} h_{\alpha_j}, \quad i = 1,\ldots,\ell.$$

This leads again to maximum two updates for each $\beta \in \Phi$ when we consider $[e_\beta, h_i],\ i = 1,\ldots,\ell.$

The cost analysis for B_ℓ is almost identical to that of C_ℓ, we obtain precisely the same number of updates. We find again that for all $\alpha \in \Phi$, $\mathrm{ad}_{e_\alpha}^{k} = 0$ whenever $k > 2$, so that Proposition 3 holds with $k_* = 2$. And $\mathrm{rank}(\mathrm{ad}_{e_\alpha}^{2})$ is either 1 or 3 for $\alpha \in \Phi$. The update counts for the computation of $d\Psi_u^{-1}(v)$ is identical to those in Table 2.

4.4 The Lie algebra D_ℓ

The Lie algebra $\mathfrak{so}(2\ell,\mathbb{C})$ can be represented as matrices in the form

$$\begin{bmatrix} Q & M \\ N & -Q^{\mathrm{T}} \end{bmatrix}, \quad M, N, Q \in \mathbb{C}^{\ell\times\ell},\ M^{\mathrm{T}} = -M,\ N^{\mathrm{T}} = -N.$$

Alternatively, we can use $2\ell \times 2\ell$ skew-symmetric complex matrices as representation.

The Cartan matrix is

$$\mathbf{C}_{D_\ell} = \begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & -1 & 2 & -1 & & \\ & & & -1 & 2 & -1 & -1 \\ & & & & -1 & 2 & 0 \\ & & & & -1 & 0 & 2 \end{bmatrix},$$

Table 3. Number of updates needed in the computation of $d\Psi_u^{-1}(v)$ for D_ℓ

Elements of $\mathfrak{h}$	$4\ell^2 - 4\ell$
Reflected root	$2\ell^2 - 2\ell$
$\mathrm{ad}\,^2_{u_\beta e_\beta}$	$\ell^2 - \ell$
All other roots	$4\ell^3 - 12\ell^2 + 8\ell$
Total	$4\ell^3 - 5\ell^2 + \ell$

and the positive roots are [16, p. 298]

$$\Phi^+ = \{\beta_{i,j},\ 1 \le i \le j < \ell\} \bigcup \{\beta_{i,\ell-2} + \beta_{j,\ell},\ 1 \le i < j \le \ell\},$$

where $\beta_{p,q}$ is ignored when $p > q$. The dimension of $\mathfrak{so}(2\ell, \mathbb{C})$ is thus $2\ell^2 - \ell$.

Now all roots can be written in the form $\beta = \sum_{n=i_\beta}^{j_\beta} r_{\beta,n}\alpha_n$, where at least $r_{\beta,i_\beta} \neq 0$, $r_{\beta,j_\beta} \neq 0$, and where all $r_{\beta,n}$ are either nonnegative or nonpositive. One can use the same approach as for C_ℓ to argue that if the roots are ordered according to the rules

1. $\alpha_+ \in \Phi^+,\ \alpha_- \in \Phi^- \ \Rightarrow\ \alpha_+ < \alpha_-$
2. If $\alpha = \displaystyle\sum_{n=i_\alpha}^{j_\alpha} r_{\alpha,n}\alpha_n,\ \beta = \sum_{n=i_\beta}^{j_\beta} r_{\beta,n}\alpha_n,\ \alpha, \beta \in \Phi^+\ (\Phi^-\ \text{resp.})\ \text{then}$
   $\alpha < \beta \ \Leftarrow\ j_\alpha > j_\beta,$

then the corresponding ordered basis is an AOB.

As basis for $\mathfrak{h}$, we choose $(h_1, \ldots, h_\ell)$, where

$$h_i = \sum_{j=i}^{\ell} h_{\alpha_j} - \frac{1}{2}(h_{\alpha_{\ell-1}} + h_{\alpha_\ell}),\ 1 \le i \le \ell - 1,$$
$$h_\ell = \frac{1}{2}(-h_{\alpha_{\ell-1}} + h_{\alpha_\ell}).$$

Again, we can use $k_* = 2$ in Proposition 3, and in this case $\mathrm{rank}(\mathrm{ad}\,^2_{e_\beta}) = 1$ for all $\beta \in \Phi$.

We omit the details of computational cost, the main results are given in Table 3.

4.5 Other semisimple Lie algebras

Apart from the Lie algebras A_ℓ – D_ℓ there exists only a finite number of semisimple Lie algebras with irreducible root systems, meaning that the base

Δ cannot be partitioned into two disjoint subsets which are orthogonal with respect to the Killing form (alternatively, the Cartan matrix is an irreducible matrix). In the case of Lie algebras with reducible root systems, we can divide them into irreducible subsystems and treat each of them separately.

The remaining root systems are G_2, F_4, E_6, E_7 and E_8. In principle, one can do the same type of analysis for these cases as discussed above, but we shall only give a few details for the case G_2, having Cartan matrix

$$\mathbf{C}_{G_2} = \begin{bmatrix} 2 & -1 \\ -3 & 2 \end{bmatrix}.$$

The dimension of G_2 is 14, thus there are 6 positive roots, namely

$$\alpha_2,\ \alpha_1+\alpha_2,\ 3\alpha_1+2\alpha_2,\ 2\alpha_1+\alpha_2,\ 3\alpha_1+\alpha_2,\ \alpha_1.$$

In fact, if the corresponding basis is ordered with the positive roots first as above, then followed by the negative roots in the same order and finally a basis for $\mathfrak{h}$, then an AOB results.

For G_2, one needs to use $k_* = 3$ in Proposition 3.

5 The solvable case

In a solvable Lie algebra, there exists an ascending series of ideals, $\mathfrak{g}_k$ with $\dim \mathfrak{g}_k = k$ such that $\mathfrak{g}_1 \subset \mathfrak{g}_2 \subset \cdots \subset \mathfrak{g}_d = \mathfrak{g}$. In this case, it is particularly easy to settle the existence of an AOB, for instance by letting $\mathrm{span}(e_{d-k+1}, \ldots, e_d) = \mathfrak{g}_k$. In particular, this implies that for each i, V_i^c is stable under ad_{e_i} and this implies that $P_i^* \left(\mathrm{ad}_{e_i}^k\right)^* \varepsilon_m = 0$ for all $m < i$ in Corollary 1. However, in general, this may not be the most efficient choice of basis. Keeping in mind that $[\mathfrak{g}, \mathfrak{g}]$ is nilpotent for any solvable Lie algebra, we may hope that a significant number of the ad_{e_i} are nilpotent. Even so, we cannot, in the above situation, be guaranteed the pleasant properties of the Chevalley basis used in the semisimple case, for instance that $[e_i, e_j] \in \mathrm{span}(e_r)$ for some r. For solvable Lie algebras, there may not exist any MTS. A tempting modification could therefore be to replace the MTS by a *Cartan subalgebra*, which is, by definition, a nilpotent selfnormalizing subalgebra of $\mathfrak{g}$, guaranteed to exist at least for Lie algebras over $\mathbb{C}$ or $\mathbb{R}$. But in our setting, the Cartan subalgebras have certain unfavourable properties, for instance, they are not necessarily abelian, and their elements need not be semisimple. We believe that, in general, a more satisfactory solution is obtained if a toral subalgebra can be identified. Unfortunately, such subalgebras do not necessarily exist, but we shall present an example where such an approach can be used.

5.1 The solvable Lie algebra $\mathfrak{t}(n, \mathbb{F})$

Let $\mathbb{F}$ be either $\mathbb{R}$ or $\mathbb{C}$. The Lie algebra $\mathfrak{t}(n, \mathbb{F})$ can be represented as a subalgebra of $\mathfrak{gl}(n, \mathbb{F})$, consisting of upper triangular matrices. The dimension of $\mathfrak{t}(n, \mathbb{F})$ is clearly $d = n(n+1)/2$. The diagonal matrices form a toral n-dimensional subalgebra $\mathfrak{h}$ of $\mathfrak{t}(n, \mathbb{F})$. In particular, this subalgebra is a Cartan subalgebra for $\mathfrak{t}(n, \mathbb{F})$. The root system leads to a basis of the type

$$\mathbf{e}_i \mathbf{e}_j^{\mathrm{T}}, \quad i \le j \le n.$$

This basis yields an AOB if for instance $e_{d-n+i} = \mathbf{e}_i \mathbf{e}_i^{\mathrm{T}}, i = 1, \ldots, n$, and the remaining elements are ordered such that $\mathbf{e}_{i_1} \mathbf{e}_{j_1}^{\mathrm{T}} < \mathbf{e}_{i_2} \mathbf{e}_{j_2}^{\mathrm{T}}$ whenever $i_1 < i_2$. With this basis, we find that $\mathrm{ad}_{e_r}^2 = 0$ if $r \le d - n$ and we can derive a particularly simple form of $d\Psi_u^{-1}$, namely

$$d\Psi_u^{-1} = I - \sum_{r=1}^{d-n} \mathrm{ad}_{u_r e_r} P_r,$$

where $u = \sum_{i=1}^{d} u_i e_i$.

6 Numerical experiments

With the purpose of the numerical simulations merely being to illustrate and verify the above theory, we have chosen to only consider real matrix Lie groups $G \subset \mathrm{GL}(n, \mathbb{R})$, with corresponding Lie algebra $\mathfrak{g}$, and let the manifold M be G itself. We use the obvious real realizations of the representations of A_ℓ – D_ℓ presented in Sect. 4. In the setting of Sect. 2 the Lie group acts on itself by multiplication $\Lambda(g, p) = g \cdot p$. For any element $y \in G$, we enumerate the matrix elements column by column as $y_1, \ldots, y_{n^2}$, and define in all cases A_ℓ – D_ℓ, the map $f : G \to \mathfrak{g}$ in (2) as

$$f(y) = \left(\sum_{i=1}^{d} y_i e_i \right) / \|y\|_{\mathrm{F}}, \tag{10}$$

where $d < n^2$ is the dimension of $\mathfrak{g}$. As initial condition we use the identity matrix and we let the particular Lie algebras be A_{10}, B_5, C_5 and D_5, respectively. We compare the results from the integration methods proposed in this paper with results generated by the Munthe-Kaas method [11] optimized using free Lie algebra techniques [12, p. 20]. We denote this method by MK and let A – D denote the integrators based on the four classical Lie algebras. All the numerical methods are based on the coefficients of “the classical Runge–Kutta method” [9, p. 138] of order four.

Table 4. Flops ratio between Munthe-Kaas methods and new methods such that the obtained global error is 10^{-6}

	$P_1(N_1)$	$P_1(N_2)$	$P_1(N_3)$	$P_2(N_1)$	$P_3(N_1)$
A	4.1	2.8	2.6	4.5	3.9
B	4.5	4.7	3.9	7.3	3.2
C	4.3	3.4	4.2	3.1	2.7
D	6.0	4.3	3.3	5.9	3.0

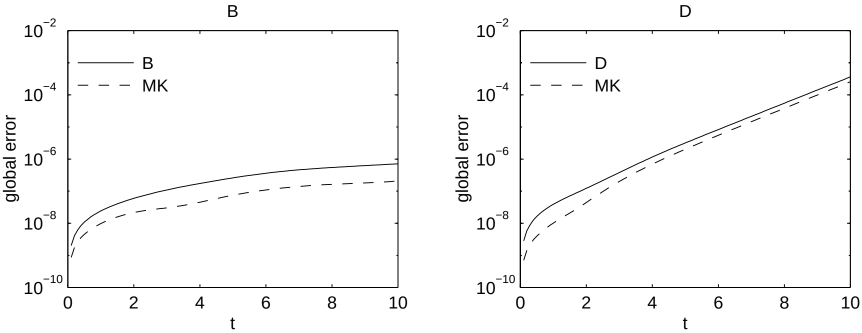

Fig. 1. Global error for simulation of (10) on the interval $[0, 10]$

Figure 1 shows global error as a function of time for B and D. For a given stepsize, the global error generated by the new methods tend to be slightly larger than the one produced by the Munthe-Kaas method, but this depends strongly on the problem. However, when measuring the efficiency as the number of flops required to obtain a prescribed global error, the proposed methods are superior to the Munthe-Kaas method, as is indicated in Figure 2 for the above problem. This observation has been supported through a number of other numerical tests we have done. Rather than giving all the details for these tests, we have chosen to present in Table 4 the ratio between the number of flops required by the Munthe-Kaas methods and the new methods in order to obtain a global error of 10^{-6} for a range of problems. The problem P_1 is defined by (10) and P_2 and P_3 are other similar artificial problems. Here $N_1 = 10,\ N_2 = 20,\ N_3 = 30$ are related to the dimension of the problem. For cases A and B, the problems are phrased in the Lie algebras $\mathfrak{sl}(N_i + 1, \mathbb{R})$ and $\mathfrak{so}(N_i + 1, \mathbb{R})$, respectively. For cases C and D, we have used $\mathfrak{sp}(N_i, \mathbb{R})$ and $\mathfrak{so}(N_i, \mathbb{R})$, respectively.

We also tested a problem based on the Lie algebra $\mathfrak{t}(n, \mathbb{R})$, defined in a similar way as (10), the difference being that we replaced $f(y)$ with $f\big((y + y^{\mathrm{T}})/2\big)$. In the case $n = 10$ we recorded a flops ratio of 5.1 for a global error of 10^{-6}.

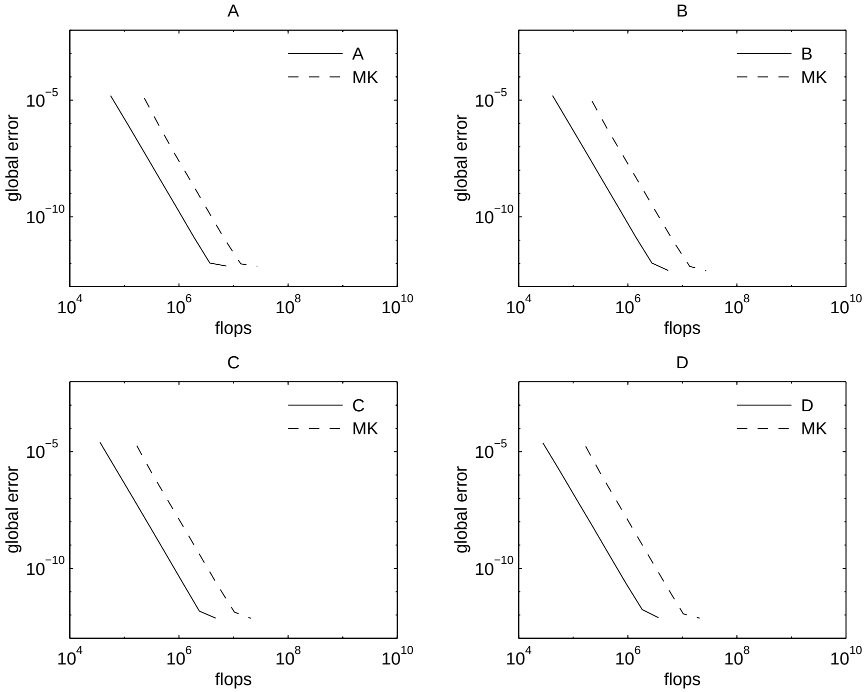

Fig. 2. Efficiency of the integrators measured as global error versus flops

We have used the MATLAB function `expm` as the exponential mapping in the Munthe-Kaas method, and the `flops` function is used for counting flops. Since the cost involved in evaluating the right hand side function f is the same for all the numerical methods, we have not included it in the flops count, the purpose being to remove factors disturbing the actual comparison of the methods.

As pointed out in Sect. 4, it is possible to count number of *updates* involved in computing $d\Psi_u^{-1}(v)$ simply by studying the properties of the underlying root systems. However, in practice there are other effects which come into play in an actual implementation. One aspect which was mentioned in Sect. 4 is that the number of flops required for each update may vary. Another factor is that there may be overhead costs involved, for instance in initializing counter variables in loops.

Table 5 shows polynomials fitted to the flop counts observed from our actual implementation of the algorithms. We see that, for B_ℓ – D_ℓ, the leading term is $8\ell^3$, which is twice the leading term in Tables 2 and 3. This is consistent with the fact that the ℓ^3 term comes from updates consisting of one multiplication and one addition.

Table 5. Computation cost measured in flops for the $d\Psi^{-1}$ mappings. The matrix representations and orderings of the basis elements are as described in Sect. 4

A_ℓ:	$\frac{5}{2}\ell^3 + 11\ell^2 + \frac{1}{2}\ell$	$(\frac{5}{2}n^3 + \frac{7}{2}n^2 - 14n + 8)$
B_ℓ:	$8\ell^3 + 8\ell^2 - 6\ell$	$(n^3 - n^2 - 4n + 4)$
C_ℓ:	$8\ell^3 + \frac{17}{2}\ell^2 - \frac{11}{2}\ell$	$(n^3 + \frac{17}{8}n^2 - \frac{11}{2}n)$
D_ℓ:	$8\ell^3 - \frac{19}{2}\ell^2 + \frac{3}{2}\ell$	$(n^3 - \frac{19}{8}n^2 + \frac{3}{4}n)$
$\mathfrak{t}(n)$:		$\frac{1}{3}n^3 + n^2 - \frac{4}{3}n$

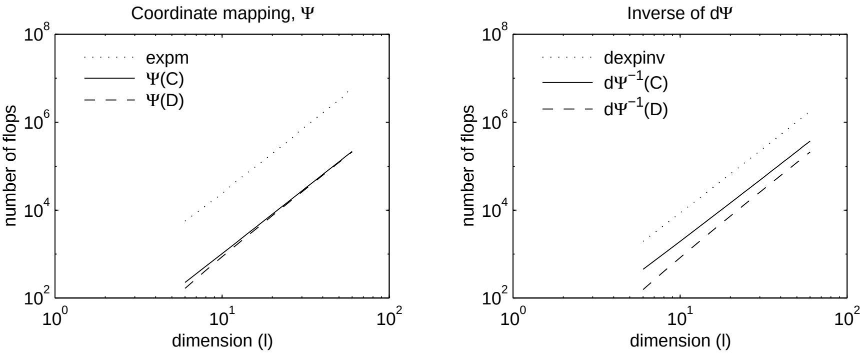

Fig. 3. Cost of computing coordinate mappings and the inverse of their differentials. The `dexpinv` function is the fourth order approximation to dexp^{-1} obtained by truncating its series

Figure 3 shows the cost of the coordinate mappings and the inverse of their differentials. The mappings $\Psi(C_\ell)$ and $\Psi(D_\ell)$ as well as $\mathrm{d}\Psi^{-1}(C_\ell)$ and $\mathrm{d}\Psi^{-1}(D_\ell)$ are computed exactly, while the exponential function is the MATLAB `expm` function. The dexp^{-1} is an infinite sum of commutators, and since the simulation in this section have been conducted with methods using coefficients of a fourth order scheme, we have let `dexpinv` be an order four approximation to dexp^{-1}.

Table 6 lists the cost involved in computing the coordinate mappings $\Psi : U \in \mathfrak{g} \to G$. Again, these polynomials are based on counting of operations in our implementation. For instance, we remark that the flops count for computing Ψ_u depends, among other things, on whether the composition of the exponentials is done from the left or right, i.e., whether one begins with $\exp(v_1 e_1)$ or $\exp(v_d e_d)$.

In developing codes for computing $d\Psi_u^{-1}$ it seems necessary to make extensive use of pointers (indirect addressing). For this reason we have not tried to analyze the cost in terms of CPU time in our numerical experiments. The efficiency with respect to time consumption will depend strongly on details of the implementation, as well as on the software and hardware

Table 6. Computation cost measured in flops for the coordinate mappings Ψ. The matrix representations and orderings of the basis elements are as described in Sect. 4

A_ℓ:	$\ell^3+2\ell^2+5\ell+4$	(n^3-n^2+4n)
B_ℓ:	$8\ell^3+24\ell^2+11\ell+1$	$(n^3+3n^2-\frac{7}{2}n+\frac{1}{2})$
C_ℓ:	$8\ell^3+\ell^2+1$	$(n^3+\frac{1}{4}n^2+1)$
D_ℓ:	$8\ell^3-5\ell^2-2\ell+1$	$(n^3-\frac{5}{4}n^2-n+1)$
$\mathfrak{t}(n)$:		$n^3-\frac{5}{4}n^2-n+1$

environment in which the code is executed. All the numerical experiments presented in this paper are done with Matlab.

There exists a number of software packages for Lie group and Lie algebra computations which may be useful tools in developing and analyzing methods of the type presented here. We mention in particular LiE [15] and GAP [14].

7 Concluding remarks and open problems

We have presented a new approach for solving ordinary differential equations on manifolds. Our main purpose has been to demonstrate that such methods can be implemented cheaply, and thereby constitute a worth while alternative to the methods of Munthe-Kaas [11]. There is still a lot of work do be done in the construction and analysis of these new methods. A natural first step will be to apply them to real problems, for instance in computational mechanics. In such settings, it has proved useful to phrase the equations and thereby the solution techniques by means of the coadjoint action of a Lie group on the dual of its Lie algebra.

In this paper, we have focused on semisimple Lie algebras $\mathfrak{g}$ over $\mathbb{C}$, but we should also be able to handle particular real forms of $\mathfrak{g}$. For instance, in the case of B_ℓ (D_ℓ resp.), it is of interest to consider the compact realization of real skew-symmetric $n\times n$ matrices where $n=2\ell+1$ ($n=2\ell$ resp.), and it is not clear from the present paper how this is done in the best way. Certainly, the natural choice of basis elements in this representation is the set of rank two matrices $\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}-\mathbf{e}_j\mathbf{e}_i^{\mathrm{T}}$, $1\leq i<j\leq n$. However, one can prove that no AOB can result from this basis if constructed in accordance with Theorem 3 when $\ell\geq 2$. Instead one can consider alternatives to AOBs. An attractive feature of the admissible ordered basis is that it causes the operator $d\Psi_u^{-1}$ to factor into d_* operators as in (6), each depending only on *one* coordinate of the second kind. We could relax on this requirement, and demand that $d\Psi_u^{-1}$ be written as a composition of somewhat fewer factors, each depending on a "small set" of coordinates, such that the cost of computing it would still be acceptably low. Another natural generalization of the present approach,

is to replace the $\exp$ function in (1) with some other map ϕ, but since the cost of computing the exponential of basis vectors of the type used in our examples is already very low, we doubt that there is much to gain in terms of computational cost by replacing $\exp$.

Apart from what can be seen in the numerical experiments, we have not been able to quantify the error growth in the new methods compared to that of the Munthe-Kaas methods. It is possible to find examples where either of the method types has the smaller global error for a fixed stepsize. However, in all the tests we have performed, the obtained global error vs the number of floating point operations has been significantly smaller with the new methods. Generally, the error growth in integration methods based on actions, is an interesting subject for further studies.

References

1. R. Carter, G. Segal, I. MacDonald. Lectures on Lie Groups and Lie Algebras, volume 32 of Student Texts. London Mathematical Society, 1995
2. E. Celledoni, A. Iserles. Approximating the exponential from a Lie algebra to a Lie group. Technical Report 1998/NA3, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1998
3. E. Celledoni, A. Iserles. Methods for the approximation of the matrix exponential in a Lie-algebraic setting. Technical Report 1999-015, MSRI, 1999. Also available as Technical Report DAMTP 1999/NA03
4. P. E. Crouch, R. Grossman. Numerical integration of ordinary differential equations on manifolds. J. Nonlinear Sci. **3**, 1–33 (1993)
5. K. Engø. On the construction of geometric integrators in the RKMK class. Technical Report No. 158, Department of Informatics, University of Bergen, 1998
6. S. Faltinsen, A. Marthinsen, H. Z. Munthe-Kaas. Multistep methods integrating ordinary differential equations on manifolds. Technical Report Numerics No. 3/1999, The Norwegian University of Science and Technology, Trondheim, Norway, 1999
7. K. Feng, Z.-J. Shang. Volume-preserving algorithms for source-free dynamical systems. Numer. Math. **71**, 451–463 (1995)
8. W. Fulton, J. Harris. Representation Theory – A First Course. Springer, Berlin, Heidelberg, New York 1991
9. E. Hairer, S. P. Nørsett, G. Wanner. Solving Ordinary Differential Equations I, Nonstiff Problems. Springer, Berlin, Heidelberg, New York, Second revised edition, 1993
10. J. E. Humphreys. Introduction to Lie Algebras and Representation Theory. Graduate Texts in Mathematics. Springer, Berlin, Heidelberg, New York, 1972
11. H. Munthe-Kaas. High order Runge–Kutta methods on manifolds. Appl. Numer. Math. **29**, 115–127 (1999)
12. H. Munthe-Kaas, B. Owren. Computations in a free Lie algebra. In C. J. Budd and A. Iserles, editors, Geometric Integration: Numerical Solution of Differential Equations on Manifolds, volume 357 of Philosophical Transactions of the Royal Society A, pages 957–982. London Mathematical Society, 1999

13. B. Owren, A. Marthinsen. Runge–Kutta methods adapted to manifolds and based on rigid frames. BIT, 39(1):116–142 (1999)
14. M. Schönert et al. GAP — Groups, Algorithms and Programming. Lehrstuhl D für Mathematik, Aachen, Germany, 3 edition, 1995
15. M. A. A. van Leeuwen. LiE, a software package for Lie group computations. Euromath Bulletin, 1(2), 1994
16. V. S. Varadarajan. Lie Groups, Lie Algebras, and Their Representations. GTM 102. Springer, Berlin, Heidelberg, New York 1984
17. A. Zanna. Collocation and relaxed collocation for the Fer and the Magnus expansions. SIAM J. Numer. Anal. 36(4):1145–1182 (1999)