Applied Numerical Mathematics 42 (2002) 465–472

APPLIED NUMERICAL MATHEMATICS

www.elsevier.com/locate/apnum

Extrapolation in Lie groups with approximated BCH-formula

Jörg Wensch [a,*], Fernando Casas [b]

[a] *MLU Halle-Wittenberg, FB Mathematik u. Informatik, Institut für Numerische Mathematik, 06099 Halle, Germany*
[b] *Departament de Matemàtiques, Universitat Jaume I, 12071 Castellón, Spain*

Abstract

We present an extrapolation algorithm for the integration of differential equations in Lie groups which is a suitable generalization of the well-known GBS-algorithm for ODEs. Sufficiently accurate approximations to the BCH-formula are required to reach a given order. We give such approximations with a minimized number of commutators. © 2002 IMACS. Published by Elsevier Science B.V. All rights reserved.

Keywords: Lie group; Geometric integration; Extrapolation methods

1. Introduction

Interest in the subject of geometric integration has been growing fast in the last years. The idea behind this subject is to join qualitative methods for dynamical systems and the computational methods of numerical mathematics. Whenever we have additional information about the structure of solutions or about the nature of the flow generated by a differential equation we want to exploit this information. We want to go back from the general purpose solver to specialized solvers for special problems.

Geometric integration includes a variety of topics from different fields of applications. We mention a few here with no claim to completeness. Crouch and Grossman propose methods for the solution of differential equations on manifolds [4]. Symplectic methods for Hamiltonian systems have been investigated by several authors: Calvo and Sanz-Serna [17], Hairer [8], Blanes et al. [1] and Yoshida [22]. Hochbruck and Lubich have constructed exponential integrators that are especially designed for the integration of highly oscillatory problems [9,10]. Several applications lead to differential equations that are naturally formulated in Lie groups. The motion of multibody systems in robotics [13] can be viewed this way. Further applications are found in molecular dynamics [19] and numerical linear algebra [5]. Calvo, Iserles and Zanna investigate methods for isospectral flows [3]. Munthe-Kaas investigates Runge–

[*] Corresponding author.
E-mail address: wensch@nummulit.mathematik.uni.halle.de (J. Wensch).

Kutta methods for the integration in Lie groups [14]. Numerical comparisons can be found in [13]. Extrapolation methods can be generalized in a different way [21] than Runge–Kutta methods. Iserles, Martinsen and Norsett construct methods for linear differential equations [11]. An excellent overview on Lie-group methods can be found in [12]. For a comprehensive introduction into the subject see [16].

Our setting is as follows: Be given a Lie group G with corresponding Lie algebra $\mathfrak{g}$ being the tangent space at the identity. A differential equation in the Lie group is given by a map $f : G \to \mathfrak{g}$. A solution of the differential equation is a curve $y(t)$ evolving in the Lie group satisfying

$$y'(t) = dR_y \circ f\bigl(y(t)\bigr), \tag{1}$$

where dR_y is the differential of right multiplication.

2. The Gragg–Bulirsch–Stoer algorithm in a Lie group

Extrapolation methods are an important subclass of one-step methods for the integration of ODEs. An extrapolation method applies a simple low order method (the underlying method) with different stepsizes $h_i = H/n_i$ to compute multiple solutions at a given point $t + H$. These solutions are combined to get a higher order approximation (extrapolation to stepsize $h = 0$). Most efficient are those methods which possess a quadratic expansion of the global error [18]. These concepts can be carried over to the Lie group setting as in [21]:

The general one-step method in a Lie group advances from an approximation y_n at time t_n to time t_{n+1} via $y_{n+1} = \exp(h\Phi(t_n, y_n, h))y_n$ where the increment function Φ maps into the Lie algebra. We have to consider both, the local error le and the global error e defined in this setting by

$$\begin{aligned} y_{n+1} &= \exp\bigl(le(t_n, y_n, h)\bigr)y(t_n + h) \quad &&\text{with } y_n = y(t_n), \text{ and}\\ y_n &= \exp\bigl(e(t_n)\bigr)y(t_n) &&\text{with } y_0 = y(t_0). \end{aligned}$$

Within this framework it can be proved [21] that:

- The global error of a one-step method of order p possesses, for sufficient smooth right-hand sides, an asymptotic expansion in powers of h, i.e., when we fix t and vary h then we have
$$e(t, h) = h^p e_p(t) + h^{p+1} e_{p+1}(t) + \cdots + \mathcal{O}\bigl(h^{N+1}\bigr).$$
- If the underlying one-step method is symmetric then this asymptotic expansion contains only even powers of h.
- The following generalization of the explicit midpoint rule
$$y_1 = \exp\bigl(hf(y_0)\bigr)y_0, \qquad y_{n+1} = \exp\bigl(2hf(y_n)\bigr)y_{n-1}, \quad n > 0, \tag{2}$$
can be interpreted as a symmetric one-step method in the even points t_{2k}. Therefore the approximations y_{2k} have a quadratic error expansion.

The explicit midpoint rule clearly is our number one candidate for an extrapolation procedure—it is explicit and symmetric, both. No Runge–Kutta method possesses this property. We choose a basic stepsize H and the stepnumber sequence $n_i = i$. For each stepnumber $i \leqslant l$ we execute $2i$ steps of the method (2) with stepsize $h_i = H/(2i)$:

$$\begin{aligned}
&Y_0^i = y_n, \qquad Y_1^i = \exp\bigl(h_i f(y_n)\bigr) \cdot y_n, \\
&Y_{k+1}^i = \exp\bigl(2h_i f\bigl(Y_k^i\bigr)\bigr) \cdot Y_{k-1}^i, \quad k = 1, \ldots, 2i-1, \\
&y_{n+1}^i := Y_{2i}^i.
\end{aligned} \tag{3}$$

Let ϕ_H be a generator of the exact flow, i.e., $y(t_{n+1}) = \exp(\phi_H) y(t_n)$ where $y(t_n) = y_n$. The results from [21] together with the BCH-formula give

$$y_{n+1}^i = \exp\bigl(\phi_H + e_2 h_i^2 + e_4 h_i^4 + \cdots\bigr) \cdot y_n. \tag{4}$$

In order to compute a higher order approximation to ϕ_H via extrapolation we have to compute $\phi_H^i \in \mathfrak{g}$ that satisfies $y_{n+1}^i = \exp(\phi_H^i) y_n$. It is given exactly by

$$\exp\bigl(\phi_H^i\bigr) = \exp\bigl(2h_i f\bigl(Y_{2i-1}^i\bigr)\bigr) \cdot \exp\bigl(2h_i f\bigl(Y_{2i-3}^i\bigr)\bigr) \cdot \cdots \cdot \exp\bigl(2h_i f\bigl(Y_1^i\bigr)\bigr). \tag{5}$$

In our algorithm we approximate ϕ_H^i by suitable approximations to the BCH-formula.

A complete extrapolation step of fixed order $p = 2l$ with basic stepsize H is given by

```
(I)   for $i = 1, \ldots, l$ do
        (a) compute $Y_1^i, \ldots, Y_{2i-1}^i$ by (3) with $h_i = H/(2i)$
        (b) compute $\tilde{\phi}_H^i = \mathrm{BCH}(h_i f(Y_1^i), h_i f(Y_3^i), \ldots, h_i f(Y_{2i-1}^i), 2l)$
(II)  compute $\phi_{\mathrm{EXT}}$ by the standard $h^2$-extrapolation tableau with initial entries $\tilde{\phi}_H^1, \ldots, \tilde{\phi}_H^l$
(III) update $y_{n+1} = \exp(\phi_{\mathrm{EXT}}) y_n$.
```

When $G = \mathbb{R}^n$ the algorithm reduces to the classical Gragg–Bulirsch–Stoehr algorithm. Note that the extrapolation approach differs from the RK–MK approach. There the differential equation is considered as an equation in the Lie algebra [14]. An approximation of the $\mathrm{d}\exp^{-1}$ –map is needed (in spite of BCH in our case) for methods of order greater than 2. As a result there is a cost of 2 commutators for the RK–MK method of order 4 based on the classical RK method [15] whereas in our method of order 4 only one commutator has to be computed.

3. Approximations to the BCH-formula

The algorithm (I)–(III) requires to consider approximations to the BCH formula up to a given order involving the minimum number of Lie brackets. If X and Y are two elements of the Lie algebra $\mathfrak{g}$ then

$$\exp(X)\exp(Y) = \exp(Z), \tag{6}$$

where

$$Z = \sum_{i>0} C_i(X, Y) \tag{7}$$

is given by the famous Baker–Campbell–Hausdorff formula [20]. Here $C_i(X, Y)$ lies in the free Lie algebra $\mathcal{G}(X, Y)$ generated by $\{X, Y\}$. Specifically, it is a linear combination of iterated commutators $[X_1, X_2, \ldots, X_i]$ where $X_j \in \{X, Y\}$. Because $\exp(-Z) = \exp(-Y)\exp(-X)$ we have the following symmetries:

$$\begin{aligned} C_i(-X,-Y) &= (-1)^i C_i(X,Y),\\ C_i(X,Y) &= (-i)^{i+1} C_i(Y,X). \end{aligned} \tag{8}$$

In the particular case which we are interested in, i.e., when X and Y are sufficiently differentiable in h, $X = \mathrm{O}(h)$, $Y = \mathrm{O}(h)$ and $Y - X = \mathrm{O}(h^2)$ we can state the following

Theorem 1. *For every $m \geqslant 1$ it is true that $C_{2m}(X,Y) = \mathcal{O}(h^{2m+1})$ and $C_{2m+1}(X,Y) = \mathcal{O}(h^{2m+3})$.*

Proof. Each function C_i can be written in the form $C_i(X,Y) = F_i(X,Y) - (-1)^i F_i(Y,X)$, where F_i is a linear combination of terms with $i-1$ nested commutators in the free Lie algebra $\mathcal{G}(X,Y)$.

Our first claim is that $F_i(X,Y) = \mathrm{O}(h^{i+1})$ for all $i > 2$. In fact, F_i is a linear combination of terms of the form

$$V := \bigl[U_1, \bigl[U_2, \ldots, \bigl[U_{i-2}, [U_{i-1}, U_i]\bigr]\cdots\bigr]\bigr],$$

where $U_k \in \{X, Y\}$. For every nonzero term of this form we may assume without loss of generality that $U_{i-1} = X$, $U_i = Y$. But $[X,Y] = [X, Y - X] = \mathrm{O}(h^3)$. Since $U_k = \mathrm{O}(h)$ for $k = 1, 2, \ldots, i-2$, we deduce our claim by adding the powers of h.

Assuming that X and Y are sufficiently differentiate in h we can expand F_i in powers of h. The condition $Y - X = \mathrm{O}(h^2)$ implies an expansion of the form $X = Ph + Qh^2 + \cdots$, $Y = Ph + Rh^2 + \cdots$. Therefore the $\mathrm{O}(h^{i+1})$ in V is exactly

$$V_0 := h^{i+1}\bigl[P, \bigl[P, \ldots, [P, Q]\cdots\bigr]\bigr].$$

Note further that swapping X and Y, the term V_0 is replaced by $-V_0$: all the Ps remain the same, while Q is replaced with $-Q$. Therefore, the $\mathrm{O}(h^{2m+2})$ terms in $C_{2m+1}(X,Y) = F_{2m+1}(X,Y) + F_{2m+1}(Y,X)$ disappear and the proof is complete. □

For an approximation of order $p = 2l$ we have to take into account iterated commutators with at most $p-2$ entries. This gives as a first result

$$\mathrm{BCH}(X,Y,4) = X + Y + \frac{1}{2}[X,Y]. \tag{9}$$

To obtain higher order approximations we apply the optimization technique presented in [2], which is formulated in the general framework of a graded free Lie algebra $\mathcal{G}(\mathcal{B})$ generated by a set of operators $\mathcal{B}$ [15], as follows. Given an element $Z \in \mathcal{G}(\mathcal{B})$ of the form

$$Z = \sum_{i=1}^{p} \sum_{j=1}^{\nu_i} \alpha_{i,j} X_{i,j},$$

where $X_{i,j}$ denotes the jth element of $\mathcal{G}(\mathcal{B})$ of grade i. In [2] a procedure is designed to obtain an approximate expression for Z up to grade p involving, in principle, the minimum number of commutators.

In the particular case of $\mathcal{B} = \{X, Y\}$ and $Z = \mathrm{BCH}(X,Y)$ this technique leads to the following results.

Order 6, 2 exponentials. We have to consider terms up to $i = 4$ in (7). The corresponding expression can be written exactly in terms of only 3 commutators as

$$\mathrm{BCH}(X,Y,6) = X + Y + \frac{1}{2}d_2 + \frac{1}{4}d_3 \tag{10}$$

with

$$d_1 = [X, Y], \qquad d_2 = \left[X + \frac{1}{6}d_1, Y\right], \qquad d_3 = \left[X, -\frac{2}{3}d_1 + d_2\right]. \tag{11}$$

Order 8, 2 exponentials. We cannot solve simultaneously all the equations required to express Z up to order 8 in terms of 5 commutators. With one additional commutator we have the following incremental scheme:

$$\mathrm{BCH}(X, Y, 8) = X + Y + \sum_{i=1}^{5} \beta_i d_i + \beta_6 [d_3, d_4], \tag{12}$$

where d_1, d_2, d_3 are given by (11),

$$\begin{aligned} d_4 &= \frac{1}{36}[X + \alpha_1 Y + \alpha_2 d_2 + \alpha_3 d_3, 4d_1 - 6d_2 - 3d_3], \\ d_5 &= \left[X + x_{5,1} Y + \sum_{i=2}^{4} x_{5,i} d_i, \sum_{i=1}^{4} y_{5,i} d_i\right], \end{aligned} \tag{13}$$

and the exact values of the coefficients are

$$\begin{aligned}
&\alpha_1 = 2 + \sqrt{3}, \qquad \alpha_2 = -\frac{9(4 + 5\sqrt{3})}{118}, \qquad \alpha_3 = -\frac{3(110 + 49\sqrt{3})}{236}, \\
&x_{5,1} = 2 - \sqrt{3}, \qquad x_{5,2} = -\frac{3(586 + 231\sqrt{3})}{1534}, \\
&x_{5,3} = -\frac{3(-17972 + 27331\sqrt{3})}{181012}, \qquad x_{5,4} = -\frac{9(23707 + 4721\sqrt{3})}{90506}, \\
&y_{5,1} = \frac{4 - \sqrt{3}}{9}, \qquad y_{5,2} = \frac{1 + \sqrt{3}}{6}, \qquad y_{5,3} = \frac{1 + \sqrt{3}}{12}, \qquad y_{5,4} = 1, \\
&\beta_1 = \frac{-9 + 5\sqrt{3}}{30}, \qquad \beta_2 = \frac{4}{5} - \frac{1}{2\sqrt{3}}, \qquad \beta_3 = \frac{3}{20}, \\
&\beta_4 = \frac{-1 + \sqrt{3}}{20}, \qquad \beta_5 = \frac{1}{20}, \qquad \beta_6 = -\frac{21(-32 + 19\sqrt{3})}{1180}.
\end{aligned}$$

Order 6, 3 exponentials. To compute $Z = \mathrm{BCH}(X_1, X_2, X_3, 6)$ with $X_i = \mathrm{O}(h)$ and $X_i - X_j = \mathrm{O}(h^2)$ we first introduce the operators $b_1 = X_1$, $b_2 = X_2 - X_1$, $b_3 = X_3 - X_1$, then we express Z in terms of b_1, b_2, b_3 by successive application of the BCH formula and finally apply the optimization technique of [2] to Z as an element of the graded free Lie algebra generated by $\{b_1, b_2, b_3\}$. As a result we can write

$$\mathrm{BCH}(X_1, X_2, X_3, 6) = X_1 + X_2 + X_3 + d_3 - d_4, \tag{14}$$

where

$$\begin{aligned}
d_1 &= \left[X_1 - \frac{13}{12}X_2, \frac{11}{13}X_2 - \frac{12}{13}X_3\right], \\
d_2 &= \left[X_1 - \frac{13}{11}X_3 - \frac{1339}{704}d_1, \frac{11}{824}X_2 + \frac{7}{6592}X_3 - \frac{1053}{8192}d_1\right],
\end{aligned}$$

$$
\begin{aligned}
d_3 &= \left[X_1 - X_3 - \frac{3965}{1236}d_1 - \frac{8}{3}d_2, X_2 + X_3 - \frac{164957}{9888}d_1 + \frac{5}{3}d_2\right], \\
M_1 &= X_1 - X_3 - \frac{2561}{309}d_1 + \frac{752}{3}d_2 - 2d_3, \\
M_2 &= \frac{1}{2}X_2 + \frac{1}{2}X_3 - \frac{160745}{9888}d_1 - \frac{179}{3}d_2 + \frac{3}{8}d_3, \\
d_4 &= [M_1, M_2].
\end{aligned}
$$

This approximation requires the minimum number of 4 commutators.

4. Numerical experiments

As a test example we choose an equation that originates from [23]. The right-hand side $f : G \to \mathfrak{g}$ is simply given in MATLAB notation by

$$f(x) = \operatorname{diag}\big(\operatorname{diag}(x, +1), +1\big) - \operatorname{diag}\big(\operatorname{diag}(x, +1), -1\big).$$

The solution of this differential equation evolves in the Lie group $SO(5)$, the group of orthogonal matrices of dimension 5. The Lie algebra of $SO(5)$ is just the set of skew-symmetric matrices of dimension 5.

We have implemented the extrapolation procedure (I)–(III) with fixed order 4 (GBSG4) and 6 (GBSG6) in MATLAB. The method of order 4 requires 4 exponentiations and 1 commutator whereas the method of order 6 requires 8 exponentiations and 7 commutators.

For comparison we use a RK–MK method of order 4 based on the classical Runge–Kutta method. This method requires 4 exponentiations and 2 commutators, see [15].

Further, for flows in $SO(n)$ Cayley methods can be applied. These methods are based on the Cayley transform $Q = (I - A)^{-1}(I + A)$ where A is skew symmetric and Q is orthogonal. For details see [6,7]. The 4th order method CAY4 is based on the classical Runge–Kutta method whereas the 6th order method CAY6 is based on extrapolation with the explicit midpoint rule.

All methods have been applied with fixed stepsize.

The left picture in Fig. 1 displays the error in the endpoint against the stepsize. Among the methods of order 4 the geometric GBS seems to perform slightly better than the other methods whereas the methods of order 6 are almost comparable.

We must not forget that the Cayley methods are especially designed for problems in $SO(n)$ while the RKMK method and our extrapolation method works in arbitrary Lie groups. It is therefore not surprising that we get a different picture when we plot error against the number of floating point operations as in the right picture in 1.

Among the methods of order 4 the extrapolation procedure is almost as efficient as the Cayley method for high accuracy. For low accuracy the Cayley method is significantly more efficient than both Lie group methods of order 4. The most efficient method for this example is the Cayley method of order 6. Our extrapolation procedure of order 6 is costly, especially for low up to moderate accuracies. The reason is the high cost for the computation of matrix exponentials.

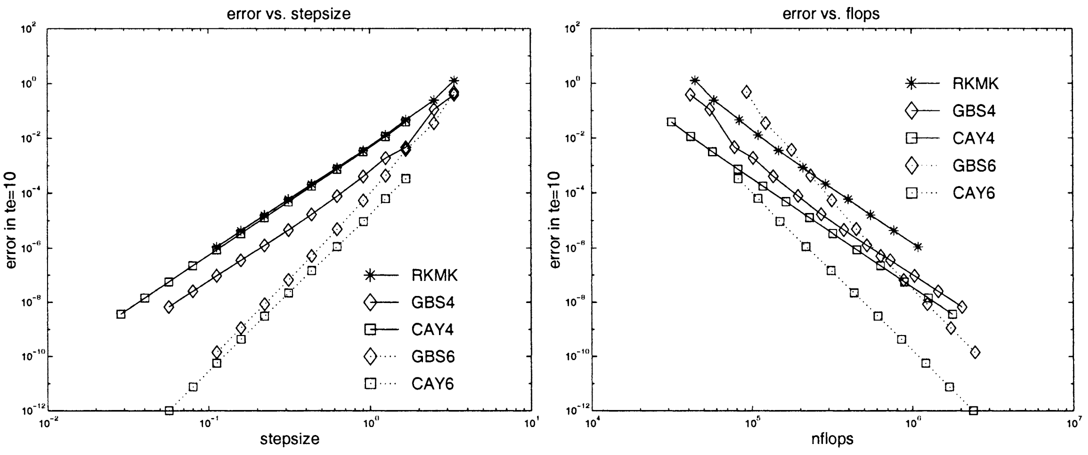

Fig. 1. Error in endpoint of solution vs. stepsize and computational effort.

5. Open questions

Modern extrapolation codes use stepsize control and order control. For that reason we need incremental BCH formulas where we can reuse the commutators in the lower order approximation for the higher order approximation.

In the case of two exponentials we succeeded in giving an incremental formula for order 8 using only 3 additional commutators besides the 3 commutators computed for the order 6 approximation. A lower bound for the required number of commutators for an order 8 approximation is 5 but we strongly believe that there are 6 commutators required. Note that for four exponentials we can utilize the formula for two exponentials:

$$\mathrm{BCH}(X_1, X_2, X_3, X_4, p) = \mathrm{BCH}\big(\mathrm{BCH}(X_1, X_2, p), \mathrm{BCH}(X_3, X_4, p), p\big). \tag{15}$$

This procedure does not work efficiently for three exponentials because of $\mathrm{BCH}(X_1, X_2) - X_3 \neq \mathcal{O}(h^2)$. The computation of an efficient formula of order 8 for 3 exponentials remains an open question.

Acknowledgements

We thank the referee for simplifying the proof of Theorem 1.

References

[1] S. Blanes, F. Casas, J. Ros, Symplectic integration with processing: A general study, SIAM J. Sci. Comput. 21 (2) (1999) 711–727.

[2] S. Blanes, F. Casas, J. Ros, High order optimized geometric integrators for linear differential equations, Preprint GIPS 2001-004, 2001.

[3] M.P. Calvo, A. Iserles, A. Zanna, Numerical solution of isospectral flows, Math Comp. 66 (1997) 1461–1486.

[4] P.E. Crouch, R. Grossman, Numerical integration of ordinary differential equations on manifolds, J. Nonlinear Sci. 3 (1) (1993) 1–33.

[5] J.W. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, PA, 1997.
[6] F. Diele, L. Lopez, R. Peluso, The Cayley transform in the numerical solution of unitary differential systems, Adv. Comput. Math. 8 (4) (1998) 317–334.
[7] F. Diele, L. Lopez, T. Politi, One step semi-explicit methods based on the Cayley transform for solving isospectral flows, J. Comput. Appl. Math. 89 (1998) 219–223.
[8] E. Hairer, Backward analysis of numerical integrators and symplectic methods, Ann. Numer. Math. 1 (1–4) (1994) 107–132.
[9] M. Hochbruck, C. Lubich, Exponential integrators for quantum-classical molecular dynamics, BIT 39 (4) (1999) 620–645.
[10] M. Hochbruck, C. Lubich, Exponential integrators for large systems of differential equations, SIAM J. Sci. Comput. 19 (5) (1998) 1552–1574.
[11] A. Iserles, A. Marthinsen, S.P. Nørsett, On the implementation of the method of Magnus series for linear differential equations, Technical Report 1998/NA02, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK.
[12] A. Iserles, H. Munthe-Kaas, S.P. Norsett, A. Zanna, Lie-group methods, Acta Numerica (2000) 215–365.
[13] A. Marthinsen, H. Munthe-Kaas, B. Owren, Simulation of ordinary differential equations on manifolds—Some numerical experiments and verifications, in: Applied Modelling and Simulation—Proceedings of the 38th SIMS Simulation Conference, Tapir trykkeri, Norway, 1996.
[14] H. Munthe-Kaas, High order Runge–Kutta methods on manifolds, J. Appl. Numer. Math. 29 (1999) 115–127.
[15] H. Munthe-Kaas, B. Owren, Computations in a free Lie Algebra, Philos. Trans. Roy. Soc. A 357 (1999) 957–981.
[16] P.J. Olver, Applications of Lie Groups to Differential Equations, Graduate Texts in Math., Vol. 107, Springer, Berlin, 1986.
[17] J.M. Sanz-Serna, M.P. Calvo, Numerical Hamiltonian Problems, Chapman & Hall, London, 1994.
[18] H.J. Stetter, Symmetric two-step algorithms for ordinary differential equations, Computing 5 (1970).
[19] M. Toda, Theory of Nonlinear Lattices, Springer, Berlin, 1978.
[20] V.S. Varadarajan, Lie Groups, Lie Algebras and their Representations, Springer, Berlin, 1984.
[21] J. Wensch, Extrapolation methods in Lie groups, Numerische Mathematik, DOI 10.1007/S002110000230.
[22] H. Yoshida, Construction of higher order symplectic integrators, Phys. Lett. A 150.
[23] A. Zanna, The method of iterated commutators for ordinary differential equations on Lie groups, Technical report 1996/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK, 1996.