Methods for the approximation of the matrix exponential in a Lie-algebraic setting

Elena Celledoni,*
MSRI,
1000 Centennial Drive, Berkeley CA 94720,
celledon@msri.org,

Arieh Iserles,
DAMTP,
Cambridge University,
Silver Street, CB3 9EW,
Cambridge, England
ai@damtp.cam.ac.uk.

February 25, 1999

Abstract

Discretization methods for ordinary differential equations based on the use of matrix exponentials have been known for decades. This set of ideas has come off age and acquired greater urgency recently, within the context of *geometric integration* and discretization methods on manifolds based on the use of Lie-group actions.

In the present paper we study the approximation of the matrix exponential in a particular context: given a Lie group G and its Lie algebra $\mathfrak{g}$, we seek approximants $F(tB)$ of $\exp(tB)$ such that $F(tB) \in \mathrm{G}$ if $B \in \mathfrak{g}$. Having fixed a basis $V_1, \ldots, V_d$ of $\mathfrak{g}$, we write $F(tB)$ as a composition of exponentials of the type $\exp(\alpha_i(t)V_i)$, where α_i for $i = 1, 2, \ldots, d$ are scalar functions. In this manner it becomes possible to increase the order of the approximation without increasing the number of exponentials to evaluate and multiply together. We study order conditions and implementation details and conclude the paper with some numerical experiments.

1 Introduction

Although numerical methods for the integration of ordinary differential equations (ODEs) based on the use of the matrix exponential have long history, the subject has acquired new relevance recently with two developments. The first, which is irrelevant to the theme of this paper, is the introduction of Krylov subspace techniques and their application to large stiff systems of differential equations (Hochbruck, Lubich & Selhofer 1998). The other development is motivated by the philosophy of *geometric integration* and its purpose is to recover under discretization important qualitative and geometric features of the underlying dynamical system. Examples of such methods can be found *inter alia* in (Casas 1996, Crouch & Grossman 1993). An important technique in geometric integration is the use of Lie-group actions, which lend themselves to the design of very effective time-stepping methods for ODEs evolving on homogeneous manifolds. Such methods have been recently studied in (Munthe-Kaas 1997) and (Engø 1998). Methods based on the use of the classical

*Research at MSRI is supported in part by NSF grant DMS-9701755.

Magnus and Fer expansions for integrating ODEs on Lie-groups can be brought into this formalism (Iserles & Nørsett 1997, Iserles & Nørsett 1999, Zanna 1997). All such methods require a repeated evaluation of a matrix exponential, often of large matrices. Inasmuch as typically one can expect the replacement of the exact exponential by a suitable approximant (a rational function, say, a Krylov subspace approximant or a Schur factorization), the context of Lie-group methods imposes a crucial extra requirement. The approximant in question, applied to an arbitrary element of the Lie algebra $\mathfrak{g}$, must produce an outcome in the Lie group G, otherwise the whole purpose of the calculation, dicretizing within G, will be null and void. This can be done is *some,* but by no means, all Lie algebras of interest and we refer the reader to (Celledoni & Iserles 1998) for a more substantive discussion of this issue.

Let G be a finite-dimensional Lie group. For all practical purposes, we may assume that G is a subgroup of the *general linear group* $\mathrm{GL}(n)$, the set of all nonsingular $n \times n$ matrices. We denote by $\mathfrak{g}$ the Lie algebra corresponding to G, observing that it is a subalgebra of $\mathfrak{gl}(n)$, the Lie algebra of all $n \times n$ matrices. Our concern in this paper is with differential equations that evolve on a manifold $\mathcal{M}$ subject to the action of G. For simplicity we can assume that $\mathcal{M}$ coincides with G and the action is of G on itself. The numerical solution of such differential equations can be obtained considering the *pull-back* on $\mathfrak{g}$, by means of the exponential map, of the vectorfield defining the equation. We can compute the corresponding flow by a Lie-algebra discretization method and recover the approximation of the original problem via exponentiation. Given an integration method of order p, we consider order-p approximants $F(tB)$ for $\exp(tB)$, where $B \in \mathfrak{g}$ and $t \geq 0$. We require that $F(tB) \in \mathrm{G}$, whence it is easy to prove that important qualitative features of the original equation and the order of the discretization are retained. In (Celledoni & Iserles 1998) we have introduced low-rank splitting methods for the construction of the approximant F, as the first attempt to provide a comprehensive treatment of this issue.

Although the constraint $F(tB) \in \mathrm{G}$ represents remarkable advantage in many applications, such as problems in which the conservation of invariants is at issue in numerical modelling (volume conservation in meteorology, invariance under rotations in the theory of mechanical systems and in robotics), it should not be interpreted as the sole purpose of our analysis. Our methods are relevant also for the approximation of $\exp(tB)$ in the more general setting $B \in \mathfrak{gl}(n)$. Suppose in fact that $B \in \mathfrak{gl}(n)$ and we want to approximate $\exp(tB)$. It is always possible to write B as a sum of a matrix $B_s \in \mathfrak{sl}(n)$ (the *special linear algebra* of $n \times n$ matrices with zero trace) and a diagonal matrix B_d whose nonzero entries are equal to $\delta = \mathrm{tr}\,(B)/n$. Then $B_s = B - B_d$ and $[B_s, B_d] = 0$ so that $\exp(tB) = \exp(tB_s)\exp(t\delta)$. This fact is a particular case of what is known in Lie theory as the Levi decomposition (Humphreys 1972, Varadarajan 1984). Using this decomposition of the matrix B, if necessary in tandem with some scaling and squaring technique, the approximation of $\exp(tB)$ can be always reduced to the approximation of $\exp(tB_s)$ with $B_s \in \mathfrak{sl}(n)$. As long as we can assure that our approximation of $\exp(tB_s)$ resides in $\mathrm{SL}(n)$, the outcome is an approximant $F(tB)$ of $\exp(tB)$ that shares with the exact exponential the feature that $\det F(tB) = \exp(\mathrm{tr}\, B)$.

It is possible to prove that, given a splitting $B = \sum_{i=1}^{k} B_i$, the function

$$\mathrm{e}^{\frac{1}{2}tB_1}\mathrm{e}^{\frac{1}{2}tB_2}\cdots\mathrm{e}^{\frac{1}{2}tB_{k-1}}\mathrm{e}^{tB_k}\mathrm{e}^{\frac{1}{2}tB_{k-1}}\cdots\mathrm{e}^{\frac{1}{2}tB_2}\mathrm{e}^{\frac{1}{2}tB_1},$$

known as the generalized *Strang splitting,* approximates $\exp(tB)$ to order 2. As long as $B_1, B_2, \ldots, B_k \in \mathfrak{g}$, it follows at once from the definition of a Lie group that the approximant resides in G. Moreover, $2k-1$ is the least number of exponentials that render such a splitting into a second-order approximant (Celledoni & Iserles 1998). The Strang splitting is time reversible, hence it follows readily from classical theory that the order can be raised from 2 to 4 by composing three Strang splittings with different time steps (Yoshida 1990). In that case we need to evaluate $3k$ exponentials and multiply $6k$ matrices. In the case of low-rank splittings which have been considered by Celledoni & Iserles (1998) this results in the following count of flops: $4n^3$ for order 2, $12n^3$ for order 4.

In this paper we present composition methods in which the number of exponentials k equals the dimension d of the Lie algebra. Our construction allows us to increase the order of the approximation without increasing the number of exponentials to evaluate and multiply together. Letting

$\{V_1,\ldots,V_d\}$ be a basis of $\mathfrak{g}$, we write $F(tB)$ as a composition of exponentials of the type $\exp(\alpha_i(t)V_i)$, where each $\alpha_i(t)$ for $i=1,\ldots,d$ is a scalar function. In general $d=\mathcal{O}(n^2)$, however, with an appropriate choice of the basis elements, the computation of each exponential $\exp(\alpha_i(t)V_i)$ requires $\mathcal{O}(n)$ flops, while the formation of their product adds just $2n^3+\mathcal{O}(n^2)$ flops. The challenging part of the computation is the construction of the functions $\alpha_1(t),\ldots,\alpha_d(t)$, and the cost of their calculation depends on the desired order of the approximation. Naive complexity analysis might have indicated that the total cost is growing exponentially in d as the order increases. Yet, the cost remains relatively modest for small orders and the method lends itself very well to the exploitation of sparsity in the matrix B. In the sequel we show how this approach can be turned into an efficient numerical method and we obtain algorithms of order up to 4 with a cost of $\mathcal{O}(n^3)$ for dense matrices.

Our approach can be interpreted as representing the solution using *canonical coordinates of the second kind,* an approach that has been pioneered by Owren & Marthinsen (1998) in the context of general Lie-group methods. Having said this, the more restrictive framework of exponential approximants possesses a very great deal of special structure. This can be exploited so as to produce efficient and competitive algorithms that approximate $\exp(tB)$, $B\in\mathfrak{g}$, in the Lie group G.

2 The technique of coordinates of second kind for the approximation of the exponential matrix

Let G be a Lie group and $\mathfrak{g}$ its corresponding d-dimensional Lie algebra. We choose a basis $\{V_1,\ldots,V_d\}$ of $\mathfrak{g}$, whence every element $Y\in \mathrm{G}$ sufficiently close to the identity can be represented in a unique fashion as

$$Y=\exp(\gamma_1V_1)\exp(\gamma_2V_2)\cdots\exp(\gamma_dV_d),$$

where $\exp:\mathfrak{g}\to\mathrm{G}$ is the exponential map. This representation is known as representation in canonical coordinates of the second kind (Varadarajan 1984). This representation is global in the case of solvable Lie algebras. We restrict ourselves to the case $\mathfrak{g}\subseteq\mathfrak{gl}(n)$, $\mathrm{G}\subseteq\mathrm{GL}(n)$, when exp is the usual matrix exponential.

Given $B\in\mathfrak{g}$, we can represent it in a unique fashion as

$$B=\sum_{i=1}^{d}\beta_iV_i.$$

It is possible then to write $\exp(tB)$ in the form

$$U(t)=\exp(tB)=\exp(g_1(t)V_1)\exp(g_2(t)V_2)\cdots\exp(g_d(t)V_d).$$

Letting $\boldsymbol{g}=[g_1,\ldots,g_d]^{\mathrm{T}}$, $\boldsymbol{\beta}=[\beta_1,\ldots,\beta_d]^{\mathrm{T}}$, it can be proved that the vector function $\boldsymbol{g}$ obeys a differential equation of the form

$$\frac{d\boldsymbol{g}}{dt}=\boldsymbol{f}(\boldsymbol{\beta},\boldsymbol{g}),\qquad \boldsymbol{g}(0)=0,$$

where $\boldsymbol{f}$ is a suitable function of $\boldsymbol{\beta}$ and $\boldsymbol{g}$, for sufficiently small t (Wei & Norman 1963). Given a solvable Lie algebra $\mathfrak{g}$ Wei & Norman (1963) prove results on the global representation of U. However, an explicit form of $\boldsymbol{f}$ is known only for very simple examples of low-dimensional Lie algebras.

In this paper we seek polynomials $\alpha_1\approx g_1,\ldots,\alpha_d\approx g_d$ of a suitable degree so that

$$\exp(tB)\approx\exp(\alpha_1(t)V_1)\exp(\alpha_3(t)V_2)\cdots\exp(\alpha_d(t)V_d).$$

Differentiation yields

$$\sum_{i=1}^{d}\beta_iV_i=\sum_{i=1}^{d}g_i'(t)\prod_{j=1}^{i-1}\mathrm{e}^{g_jV_j}V_i\prod_{j=i-1}^{1}\mathrm{e}^{-g_jV_j}. \tag{2.1}$$

Evaluating this expression at the origin gives the first-order condition

$$g_i'(0) = \beta_i, \qquad i = 1, 2, \ldots, d. \tag{2.2}$$

Further differentiations of (2.1) lead to higher-order conditions. Let us define the functions

$$P_i(\boldsymbol{g}) = \exp(\mathrm{ad}_{g_1 V_1}) \circ \cdots \circ \exp(\mathrm{ad}_{g_{i-1} V_{i-1}})(V_i), \qquad i = 1, 2, \ldots, d, \tag{2.3}$$

where the *adjoint operator* $\mathrm{ad}_x : \mathfrak{g} \to \mathfrak{g}$ is defined as $\mathrm{ad}_x(y) = [x, y]$ for any $x, y \in \mathfrak{g}$, $[x, y] = xy - yx$ being the matrix commutator. Note that $P_i(\boldsymbol{g}(0)) = V_i$, $i = 1, 2, \ldots, d$. Moreover, the right-hand side of (2.1) can be written in the simplified form The function

$$\mathcal{T}(\boldsymbol{g}) = \sum_{i=1}^{d} g_i'(t) P_i(g).$$

Since the derivatives of the left hand side of (2.1) vanish, the conditions for order $p \geq 1$, can be obtained by solving the equations

$$\left.\frac{\mathrm{d}^r}{\mathrm{d}t^r} \mathcal{T}(\boldsymbol{g})\right|_{t=0} = 0, \qquad r = 1, 2, \ldots, p-1, \quad p \geq 1, \tag{2.4}$$

where

$$\frac{\mathrm{d}^r}{\mathrm{d}t^r} \mathcal{T}(\boldsymbol{g}) = \sum_{i=1}^{d} \sum_{k=1}^{r} \binom{r}{k} \frac{\mathrm{d}^{r-k+1} g_i}{\mathrm{d}t^{r-k+1}} \frac{\mathrm{d}^k P_i}{\mathrm{d}t^k}. \tag{2.5}$$

In particular,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \mathcal{T}(\boldsymbol{g}) &= \sum_{i=1}^{d} \left(g_i'' P_i + g_i' \frac{d}{dt} P_i \right), \\
\frac{\mathrm{d}^2}{\mathrm{d}t^2} \mathcal{T}(\boldsymbol{g}) &= \sum_{i=1}^{d} \left(g_i''' P_i + 2 g_i'' \frac{\mathrm{d}}{\mathrm{d}t} P_i + g_i' \frac{\mathrm{d}^2}{\mathrm{d}t^2} P_i \right), \\
\frac{\mathrm{d}^3}{\mathrm{d}t^3} \mathcal{T}(\boldsymbol{g}) &= \sum_{i=1}^{d} \left(g_i^{IV} P_i + 3 g_i''' \frac{\mathrm{d}}{\mathrm{d}t} P_i + 3 g_i'' \frac{\mathrm{d}^2}{\mathrm{d}t^2} P_i + g_i' \frac{\mathrm{d}^3}{\mathrm{d}t^3} P_i \right).
\end{aligned}$$

Solving (2.4) for $r = 1$ results in the values of $g_i''(0)$ for $i = 1, 2, \ldots, d$ that allow us to construct an order-2 approximant. Substituting such values in (2.4) for $r = 2$ yields $g_i'''(0)$ for $i = 1, 2, \ldots, d$ and consequently an approximant of order 3. Similar procedure can be used to construct recursively approximants of arbitrarily high order.

The main part of the computation is the evaluation of the k-th derivative of $P_i(\boldsymbol{g})$ at $t = 0$. Expanding the exponentials in (2.3) we obtain

$$P_i(\boldsymbol{g}) = \prod_{k=1}^{i-1} (I + \mathrm{ad}_{g_k V_k} + \tfrac{1}{2} \mathrm{ad}_{g_k V_k}^2 + \tfrac{1}{6} \mathrm{ad}_{g_k V_k}^3 + \ldots)(V_i)$$

and, after further algebra,

$$\begin{aligned}
P_i(\boldsymbol{g}) = \Bigg\{ I &+ \sum_{k=1}^{i-1} \mathrm{ad}_{g_k V_k} + \sum_{k=2}^{i-1} \sum_{l=1}^{k-1} \mathrm{ad}_{g_l V_l} \mathrm{ad}_{g_k V_k} \\
&+ \tfrac{1}{2} \sum_{k=1}^{i-1} \mathrm{ad}_{g_k V_k}^2 + \sum_{k=3}^{i-1} \sum_{l=2}^{k-1} \sum_{j=1}^{l-1} \mathrm{ad}_{g_j V_j} \mathrm{ad}_{g_l V_l} \mathrm{ad}_{g_k V_k} \\
&+ \tfrac{1}{2} \sum_{k=2}^{i-1} \sum_{l=1}^{k-1} \left(\mathrm{ad}_{g_l V_l} \mathrm{ad}_{g_k V_k}^2 + \mathrm{ad}_{g_l V_l}^2 \mathrm{ad}_{g_k V_k} \right) + \tfrac{1}{6} \sum_{k=1}^{i-1} \mathrm{ad}_{g_k V_k}^3 + \ldots \Bigg\} (V_i).
\end{aligned}$$

Similarly to (Owren & Marthinsen 1997), we write $P_i(\boldsymbol{g})$ in the form

$$P_i(\boldsymbol{g}) = I + \sum_{r=1}^{\infty} \sum_{j_1=1}^{i-1} \sum_{j_2=j_1}^{i-1} \cdots \sum_{j_r=j_{r-1}}^{i-1} \frac{1}{\boldsymbol{j}!} g_{j_1} \dots g_{j_r} \mathrm{ad}_{V_{j_1}} \circ \cdots \circ \mathrm{ad}_{V_{j_r}}(V_i).$$

Here $\boldsymbol{j} = (j_1, \dots j_r)$ is a multi-index of integer elements with $1 \leq j_r \leq i-1$ and $\boldsymbol{j}! := q_1!q_2! \dots q_{i-1}!$ where q_k is the number of occurrences of k in $(j_1, j_2 \dots j_r)$.

A general expression for the k-th derivative of P_i is given as follows: since $g_i(0) = 0$, we may let $f_i(t) = g_i(t)/t$, $i = 1, 2, \dots, d$. We can then rewrite P_i in the form

$$P_i(\boldsymbol{g}) = I + \sum_{r=1}^{\infty} t^r \sum_{j_1=1}^{i-1} \sum_{j_2=j_1}^{i-1} \cdots \sum_{j_r=j_{r-1}}^{i-1} \frac{1}{\boldsymbol{j}!} f_{j_1} \cdots f_{j_r} \mathrm{ad}_{V_{j_1}} \circ \cdots \circ \mathrm{ad}_{V_{j_r}}(V_i).$$

By following the construction in(Owren & Marthinsen 1997) we obtain

$$\begin{aligned} \left.\frac{\mathrm{d}^k P_i}{\mathrm{d}t^k}\right|_{t=0} = \sum_{r=1}^{k} \sum_{\delta_1+\ldots+\delta_r=k} \frac{k!}{\prod_{\nu=1}^{\mu}(\delta_\nu - 1)!} \sum_{1\leq j_1 \leq \ldots \leq j_\mu \leq i-1} \frac{1}{\boldsymbol{j}!} \\ \times \left. f_{j_1}^{(\delta_1-1)} \cdots f_{j_\mu}^{(\delta_\mu-1)} \right|_{t=0} \mathrm{ad}_{V_{j_1}} \circ \cdots \circ \mathrm{ad}_{V_{j_\mu}}(V_i). \end{aligned} \tag{2.6}$$

Substituting (2.6) in (2.4) and (2.5) we obtain the conditions for arbitrary order p. In particular we obtain the following formulae for the derivatives of $P_i(\boldsymbol{g})$ at $t = 0$,

$$\begin{aligned} \left.\frac{\mathrm{d}P_i}{\mathrm{d}t}\right|_{t=0} &= \sum_{k=1}^{i-1} \mathrm{ad}_{V_k}(V_i)\, g_k'(0), \\ \left.\frac{\mathrm{d}^2 P_i}{\mathrm{d}t^2}\right|_{t=0} &= \sum_{k=1}^{i-1} \left(\sum_{l=1}^{k-1} 2\mathrm{ad}_{V_l}\mathrm{ad}_{V_k}(V_i) g_k'(0) g_l'(0) + \mathrm{ad}_{V_k}^2(V_i)[g_k'(0)]^2 + \mathrm{ad}_{V_k}(V_i) g_k''(0) \right), \\ \left.\frac{\mathrm{d}^3 P_i}{\mathrm{d}t^3}\right|_{t=0} &= 6 \sum_{k=3}^{i-1} \sum_{l=2}^{k-1} \sum_{j=1}^{l-1} \mathrm{ad}_{V_j}\mathrm{ad}_{V_l}\mathrm{ad}_{V_k}(V_i) g_j'(0) g_l'(0) g_k'(0) \\ &\quad + 3 \sum_{k=2}^{i-1} \sum_{l=1}^{k-1} \left(\mathrm{ad}_{V_l}\mathrm{ad}_{V_k}^2(V_i) g_l'(0)[g_k'(0)]^2 + \mathrm{ad}_{V_l}^2 \mathrm{ad}_{V_k}(V_i)[g_l'(0)]^2 g_k'(0) \right) \\ &\quad + \sum_{k=1}^{i-1} \mathrm{ad}_{V_k}^3(V_i)[g_k'(0)]^3 \\ &\quad + 3 \sum_{k=2}^{i-1} \sum_{l=1}^{k-1} \mathrm{ad}_{V_l}\mathrm{ad}_{V_k}(V_i) \left(g_k''(0) g_l'(0) + g_k'(0) g_l''(0) \right) \\ &\quad + 3 \sum_{k=1}^{i-1} \mathrm{ad}_{V_k}^2(V_i) g_k''(0) g_k'(0) \\ &\quad + \sum_{k=1}^{i-1} \mathrm{ad}_{V_k}(V_i) g_k'''(0). \end{aligned}$$

Substitution readily produces order conditions. Specifically,

$$\sum_{i=1}^{d} g_i''(0) V_i = -\sum_{i=1}^{d} g_i'(0) \sum_{k=1}^{i-1} \mathrm{ad}_{V_k}(V_i) g_k'(0), \tag{2.7}$$

are conditions for order 2, while

$$\begin{aligned}\sum_{i=1}^{d} g_i'''(0)V_i = -\sum_{i=1}^{d} \Bigg\{ & 2g_i''(0)\sum_{k=1}^{i-1} \mathrm{ad}_{V_k}(V_i)g_k'(0) \\ & + g_i'(0)\sum_{k=1}^{i-1}\Bigg[2\sum_{l=1}^{k-1}\mathrm{ad}_{V_l}\mathrm{ad}_{V_k}(V_i)g_k'(0)g_l'(0) \\ & + \mathrm{ad}_{V_k}^2(V_i)(g_k'(0))^2 + \mathrm{ad}_{V_k}(V_i)g_k''(0)\Bigg]\Bigg\}\,,\end{aligned} \tag{2.8}$$

are the order-3 conditions. Finally, conditions for order 4 are

$$\begin{aligned}\sum_{i=1}^{d} g_i^{\mathrm{IV}}(0)V_i = -\sum_{i=1}^{d}\Bigg\{ & 3g_i'''(0)\sum_{k=1}^{i-1}\mathrm{ad}_{V_k}(V_i)g_k'(0)+ \\ & + 3g_i''(0)\Bigg[\sum_{k=1}^{i-1}\Bigg(\sum_{l=1}^{k-1} 2\mathrm{ad}_{V_l}\mathrm{ad}_{V_k}(V_i)g_k'(0)g_l'(0) \\ & + \mathrm{ad}_{V_k}^2(V_i)(g_k'(0))^2 + \mathrm{ad}_{V_k}(V_i)g_k''(0)\Bigg)\Bigg] \\ & + g_i'(0)\Bigg[6\sum_{k=3}^{i-1}\sum_{l=2}^{k-1}\sum_{j=1}^{l-1}\mathrm{ad}_{V_j}\mathrm{ad}_{V_l}\mathrm{ad}_{V_k}(V_i)g_j'(0)g_l'(0)g_k'(0) \\ & + 3\sum_{k=2}^{i-1}\sum_{l=1}^{k-1}\left(\mathrm{ad}_{V_l}\mathrm{ad}_{V_k}^2(V_i)g_l'(0)(g_k'(0))^2 + \mathrm{ad}_{V_l}^2\mathrm{ad}_{V_k}(V_i)(g_l'(0))^2 g_k'(0)\right) \\ & + \sum_{k=1}^{i-1}\mathrm{ad}_{V_k}^3(V_i)(g_k'(0))^3 \\ & + 3\sum_{k=2}^{i-1}\sum_{l=1}^{k-1}\mathrm{ad}_{V_l}\mathrm{ad}_{V_k}(V_i)\left(g_k''(0)g_l'(0) + g_k'(0)g_l''(0)\right) \\ & + 3\sum_{k=1}^{i-1}\mathrm{ad}_{V_k}^2(V_i)g_k''(0)g_k'(0) \\ & + \sum_{k=1}^{i-1}\mathrm{ad}_{V_k}(V_i)g_k'''(0)\Bigg]\Bigg\}.\end{aligned} \tag{2.9}$$

In Figure 1 we have plotted along the y axis the 2-norm of the error of the approximation of $\exp(tB)$ with the second-kind coordinates (SKC) methods of order ranging from 1 to 4. The values of the error are plotted against time, (along the x-axis), to logarithmic scale for matrices of $\mathfrak{sl}(5)$. The methods have been implemented using the standard basis defined in section 3.

The computation of $g''(0)$, $g'''(0)$ and $g^{\mathrm{IV}}(0)$ is obtained directly implementing the formulas (2.7), (2.8),(2.9) respectively. This implementation does not depend on the choice of the particular basis of $\mathfrak{sl}(5)$, but the number of commutators that must be computed with this approach is $\mathcal{O}(d^p)$ for $p = 2, 3, 4$. Even if we assume that the V_is are very sparse matrices and that the cost of computing each commutator is $\mathcal{O}(1)$ operations, the total cost exceeds $\mathcal{O}(n^{2p})$ flops for $p = 2, 3, 4$ where n is the dimension of the matrix. Such expense is not acceptable for a competitive method of approximation of $\exp(tB)$. Fortunately, it can be decreased a very great deal by an appropriate choice of the basis $\{V_1, V_2, \dots, V_d\}$. This is the theme of the next section.

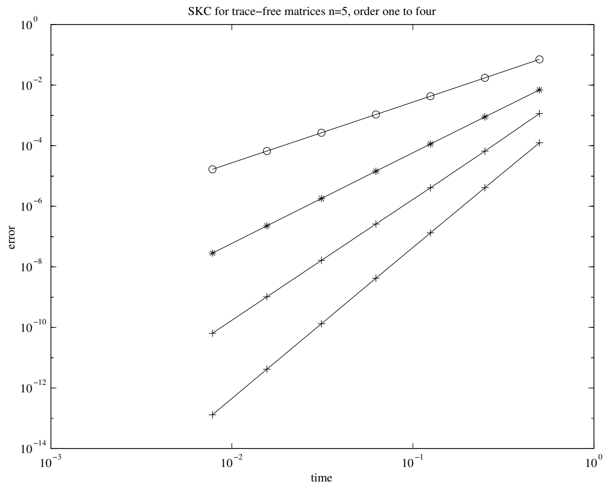

Figure 1: Error in the approximation of the exponential with WN technique.

3 Choosing a basis

The choice of the right basis and sparse representation of commutators are critical to the implementation of the SKC methods. Recalling the order conditions (2.7)–(2.9), our aim is to choose a basis so that terms of the form $\mathrm{ad}_{V_{i_1}}\mathrm{ad}_{V_{i_2}}\cdots\mathrm{ad}_{V_{i_s}}V_j$ can be represented in the most economical manner. We recall that, given the basis $\{V_1, V_2, \ldots, V_d\}$ of a d-dimensional Lie algebra $\mathfrak{g}$, the *structure constants* are the numbers $c^i_{k,l}$, $k, l, i = 1, 2, \ldots, d$, such that

$$[V_k, V_l] = \sum_{i=1}^{d} c^i_{k,l} V_i$$

(Humphreys 1972). Let

$$B = \sum_{k=1}^{d} \beta_k V_k.$$

Then an order-1 condition is always

$$g'_k(0) = \beta_k, \qquad k = 1, 2, \ldots, d. \tag{3.1}$$

To obtain the order-2 condition we substitute (3.1) in (2.7) and express commutators in terms of structure constants,

$$\begin{aligned}\sum_{k=1}^{d} g''_k(0)V_k &= -\sum_{l=1}^{d}\beta_l\sum_{j=1}^{l-1}[V_j, V_l]\beta_j = -\sum_{l=1}^{d}\beta_l\sum_{j=1}^{l-1}\sum_{k=1}^{d}c^k_{j,l}V_k \\ &= -\sum_{k=1}^{d}\left(\sum_{l=1}^{d}\sum_{j=1}^{l-1}\beta_l c^k_{j,l}\beta_j\right)V_k.\end{aligned}$$

Since $c^k_{j,l} = -c^k_{l,j}$, we thus deduce that

$$g''_k(0) = \sum_{l=1}^{d}\sum_{j=1}^{l-1}\beta_l c^k_{l,j}\beta_j, \qquad k = 1, 2, \ldots, d. \tag{3.2}$$

Likewise, substituting in (2.8),

$$\sum_{k=1}^{d} g_k'''(0)V_k = -\sum_{i=1}^{d}\left\{2g_i''(0)\sum_{l=1}^{i-1}[V_l,V_i]\beta_l + \beta_i\sum_{l=1}^{i-1}\left(2\sum_{j=1}^{l-1}[V_j,[V_l,V_i]]\beta_l\beta_j\right.\right.$$
$$\left.\left. + [V_l,[V_l,V_i]]\beta_l^2 + [V_l,V_i]g_l''(0)\right)\right\}.$$

Note that

$$[V_j,[V_l,V_i]] = \sum_{s=1}^{d} c_{l,i}^s [V_j,V_s] = \sum_{k=1}^{d}\sum_{s=1}^{d} c_{l,i}^s c_{j,s}^k V_k.$$

Therefore

$$\sum_{k=1}^{d} g_k'''(0)V_k = -2\sum_{i=1}^{d} g_i''(0)\sum_{l=1}^{i-1}\sum_{k=1}^{d} c_{l,i}^k\beta_l V_k - 2\sum_{i=1}^{d}\beta_i\sum_{l=1}^{i-1}\sum_{j=1}^{l-1}\sum_{k=1}^{d}\sum_{s=1}^{d} c_{l,i}^s c_{j,s}^k \beta_l\beta_j V_k$$
$$-\sum_{i=1}^{d}\beta_i\sum_{l=1}^{i-1}\sum_{k=1}^{d}\sum_{s=1}^{d} c_{l,i}^s c_{l,s}^k \beta_l^2 V_k - \sum_{i=1}^{d}\sum_{l=1}^{i-1}\beta_i\sum_{k=1}^{d} c_{l,i}^k g_l''(0)V_k$$

and we deduce that

$$g_k'''(0) = \sum_{i=1}^{d}\sum_{l=1}^{i-1} c_{i,l}^k[2g_i''(0)\beta_l + \beta_i g_l''(0)] + 2\sum_{i=1}^{d}\sum_{l=1}^{i-1}\sum_{j=1}^{l-1}\sum_{s=1}^{d} c_{i,l}^s c_{j,s}^k \beta_i\beta_l\beta_j$$
$$-\sum_{i=1}^{d}\sum_{i=1}^{d}\sum_{l=1}^{i-1}\sum_{s=1}^{d} c_{i,l}^s c_{l,s}^k \beta_i\beta_l^2, \qquad k=1,2,\ldots,d.$$

Bearing in mind that for order p we require

$$\alpha_k(t) = \sum_{r=1}^{p}\frac{1}{r!}g_k^{(r)}(0)t^r, \qquad k=1,2,\ldots,d,$$

we observe that the sheer volume of calculations required for the evaluation of the functions $\alpha_1,\alpha_2,\ldots,\alpha_d$ is prohibitive for, say, order 3, unless most of the structure constants vanish. Fortunately, bases of finite-dimensional Lie algebras which are 'sparse' (in the sense that a very high proportion of structure constants vanish) are known. They are associated with *root space decompositions* of Lie algebras (Humphreys 1972) and, in the case of semisimple algebras, are known as *Chevalley bases* (Carter, Segal & Macdonald 1995). Wishing to avoid too much Lie-algebraic terminology in a numerical analysis paper, we reserve our exposition to just three examples which are the most important in a range applications.

The orthogonal group Let $\mathfrak{g} = \mathfrak{so}(n)$, the Lie algebra of $n\times n$ skew-symmetric matrices. It corresponds to two important Lie groups: the orthogonal group $\mathrm{O}(n)$ of $n\times n$ orthogonal matrices and its subgroup, the special orthogonal group $\mathrm{SO}(n)$ of matrices with unit determinant. Its dimension is $d=\frac{1}{2}n(n-1)$. We let

$$F_{i,j} = \boldsymbol{e}_i\boldsymbol{e}_j^{\mathrm{T}} - \boldsymbol{e}_j\boldsymbol{e}_i^{\mathrm{T}}, \qquad i=1,2,\ldots,n, \quad j=i+1,i+2,\ldots,n,$$

where $\boldsymbol{e}_i$ is the i-th canonical vector of $\mathbb{R}^n$. In other words, $F_{i,j}$ is a matrix whose (i,j)-th element is 1, the (j,i)-th element equals -1 and zero otherwise. We can trivially expand each $B\in\mathfrak{so}(n)$ as $B=\sum_{i=1}^n\sum_{j=i+1}^n b_{i,j}F_{i,j}$. $U(t) := \exp(tF_{i,j})$ is simply an *Euler rotation* in the (i,j) plane: it is identity matrix, except that

$$\begin{bmatrix} U_{i,i} & U_{i,j} \\ U_{j,i} & U_{j,j}\end{bmatrix} = \begin{bmatrix} \cos(tF_{i,j}) & \sin(tF_{i,j}) \\ -\sin(tF_{i,j}) & \cos(tF_{i,j})\end{bmatrix}.$$

Noting that

$$[F_{i,j}, F_{l,k}] = \begin{cases} -F_{j,k}, & i = l,\ j \neq k, \\ -F_{i,l}, & i \neq l,\ j = k, \\ F_{i,k}, & i \neq k,\ j = l, \\ F_{j,l}, & i = k,\ j \neq l, \\ O, & \text{otherwise}, \end{cases}$$

the order conditions are simplified as follows,

$$\begin{aligned} p \geq 1: \qquad & g'_{i,j}(0) = b_{i,j}, \\ p \geq 2: \qquad & g''_{i,j}(0) = \sum_{s=j+1}^{n} b_{j,s} b_{i,s} - \sum_{r=i+1}^{j-1} b_{r,j} b_{i,r} \\ & \qquad + \sum_{r=1}^{i-1} b_{r,j} b_{r,i}, \qquad 1 \leq i < j \leq n, \end{aligned}$$

and similarly for higher-order terms. Thus, the cost of computing the coefficients for the second-order method is just $\frac{1}{2}(n-2)(n-1)n \approx \frac{1}{2}n^3$ flops. In comparison, a naive computation of (3.2), without exploiting sparsity of structure constants, requires $\frac{1}{8}(n^2-n-2)(n-1)^2n^2 \approx \frac{1}{8}n^6$ flops.

A more classical composition method for $B \in \mathfrak{so}(n)$ is the *Strang splitting* which we can write in the form

$$\mathrm{e}^{tb_{1,2}F_{1,2}/2} \cdots \mathrm{e}^{tb_{n-2,n}F_{n-2,n}/2} \mathrm{e}^{tb_{n-1,n}Fn-1,n} \mathrm{e}^{tb_{n-2,n}F_{n-2,n}/2} \cdots \mathrm{e}^{tb_{1,2}F_{1,2}/2}$$

(Celledoni & Iserles 1998). It gives a second-order approximant to $\exp(tB)$ whose calculation requires $\approx 4n^3$ flops, in comparison with $\approx 3n^3$ for the second-order CSK method.

We note that, in the specific case of $\mathfrak{so}(n)$, diagonal Padé approximants to the exponential provide an alternative to our method, since they map the algebra to $\mathrm{O}(n)$. Having said this, for dense matrices B the cost of evaluating the second-order approximant $(I - \frac{1}{2}tB)^{-1}(I + \frac{1}{2}tB)$ with, say, LU factorization is $\mathcal{O}(n^3)$, comparative with our method.

The special linear group Let $\mathfrak{g} = \mathfrak{sl}(n)$, the set of $n \times n$ matrices with zero trace, whence $d = n^2 - 1$. We split the algebra in the first instance into diagonal and off-diagonal parts: in the terminology of Lie algebras, the subspace spanned by the diagonal elements is a *Cartan subalgebra* (Carter et al. 1995) or *maximal toral algebra* (Humphreys 1972) of $\mathfrak{sl}(n)$. Specifically, our basis is

$$\{E_{i,j} \,:\, i,j = 1,2,\ldots,n\ i \neq j\} \cup \{D_i \,:\, i = 1,2,\ldots,n-1\}.$$

where

$$\begin{aligned} E_{i,j} &= \boldsymbol{e}_i \boldsymbol{e_j}^{\mathrm{T}}, \qquad i,j = 1,2,\ldots,n, \quad i \neq j, \\ D_i &= \boldsymbol{e}_i \boldsymbol{e_i}^{\mathrm{T}} - \boldsymbol{e}_{i+1}\boldsymbol{e}_{i+1}^{\mathrm{T}}, \qquad i = 1,2,\ldots,n-1. \end{aligned}$$

The exponentials of $E_{i,j}$ and D_i are trivial,

$$\mathrm{e}^{tE_{i,j}} = I + tE_{i,j}, \qquad \mathrm{e}^{tD_i} = \mathrm{e}^t D_i.$$

We order the elements by taking first $E_{i,j}$, $i \neq j$, in lexicographic order, followed by $D_1, D_2, \ldots, D_{n-1}$. The commutator table is

$$[E_{i,j}, E_{r,s}] = \begin{cases} E_{i,s}, & i \neq s,\ j = r, \\ -E_{r,j}, & i = s,\ j \neq r, \\ \sum_{l=s}^{r-1} D_l, & i = s < j = r, \\ -\sum_{l=r}^{s-1} D_l, & i = s > j = r, \\ O, & \text{otherwise}, \end{cases} \qquad \begin{array}{l} i,j,r,s = 1,2,\ldots,n, \\ i \neq j, \quad r \neq s, \end{array}$$

$$
[E_{i,j}, D_r] = \begin{cases} -E_{r,j}, & i = r,\ j \neq r+1, \\ -E_{i,r+1}, & i \neq r,\ j = r+1, \\ -2E_{r,r+1}, & i = r,\ j = r+1, \\ E_{r+1,j}, & i = r+1,\ j \neq r, \\ E_{i,r}, & i \neq r+1,\ j = r, \\ 2E_{r+1,r}, & i = r+1,\ j = r, \\ O, & \text{otherwise}, \end{cases} \qquad \begin{array}{l} i,j = 1,\dots,n, \quad i \neq j, \\ r = 1,\dots,n-1, \end{array}
$$

$$
[D_i, D_j] = O, \qquad i,j = 1,2,\dots,n-1.
$$

In general, for a d-dimensional Lie algebra there are $(d-1)d^2$ structure constants. In the case of $\mathfrak{sl}(n)$ this means that up to $\approx n^6$ structure constants may be nonzero. Yet, using the above basis results in just $2(n-1)n^2 + 4(n-2)(n-1) + \frac{2}{3}(n^2-1)n \approx \frac{7}{3}n^3$ nonzero structure constants and substantial saving in the implementation of the SKC technique.

Letting

$$
\begin{aligned}
[E_{i,j}, E_{r,s}] &= \sum_{(k,l)} c^{(k,l)}_{(i,j),(k,l)} E_{k,l} + \sum_k c^{(k)}_{(i,j),(r,s)} D_k, \\
[E_{i,j}, D_r] &= \sum_{(k,l)} c_{(i,j),r} E_{k,l}
\end{aligned}
$$

(note that $[D_r, E_{i,j}] = -[E_{i,j}, D_r]$ and $[D_i, D_j] = O$) we thus have

$$
\begin{aligned}
c^{(k,l)}_{(i,j),(r,s)} &= \begin{cases} +1, & k = i,\ l = s,\ r = j,\ s \neq i, \\ -1, & k = r,\ l = j,\ r \neq j,\ s = i, \\ 0, & \text{otherwise}, \end{cases} \\
c^{k}_{(i,j),(r,s)} &= \begin{cases} +1, & i = s < j = r,\ k = s, s+1, \dots, r-1, \\ -1, & i = s > j = r,\ k = r, r+1, \dots, s-1, \\ 0, & \text{otherwise}, \end{cases} \\
c^{(k,l)}_{(i,j),r} &= \begin{cases} +1, & k = i = r+1,\ l = j,\ j \neq r \text{ or } k = i,\ l = j = r,\ i \neq r+1, \\ -1, & k = i = r,\ l = j,\ j \neq r+1 \text{ or } k = i,\ l = j = r+1,\ i \neq r, \\ +2, & k = i = r+1,\ l = j = r, \\ -2, & k = i = r,\ l = j = r+1, \\ 0, & \text{otherwise}, \end{cases} \\
c^{k}_{(i,j),r} &= c^{(k,l)}_{r,s} = c^{k}_{r,s} = 0.
\end{aligned}
$$

Letting

$$
B = \sum_{k \neq l} \beta_{k,l} E_{k,l} + \sum_k \gamma_k D_k,
$$

and ordering the pairs (k,l), $k \neq l$, in lexicographic order, we thus have

$$
\begin{aligned}
g'_{k,l}(0) &= \beta_{k,l}, \\
g'_k(0) &= \gamma_k, \\
g''_{k,l}(0) &= \sum_{(i,j) \succ (r,s)} \beta_{i,j} c^{(k,l)}_{(i,j),(r,s)} \beta_{r,s} + \sum_{(i,j),r} \beta_{i,j} c^{k}_{(i,j),r} \gamma_r \\
&= \sum_{i=1}^{k-1} \beta_{k,i}\beta_{i,l} - \sum_{i=k+1}^{n} \beta_{k,i}\beta_{i,l} + \beta_{k,l}(\gamma_{k-1} + \gamma_l - \gamma_k - \gamma_{l-1}), \\
g''_k(0) &= \sum_{(i,j) \succ (r,s)} \beta_{i,j} c^{k}_{(i,j),(r,s)} \beta_{r,s} = -\sum_{i=1}^{k} \sum_{j=k+1}^{n} \beta_{i,j}\beta_{j,i},
\end{aligned}
$$

where $\gamma_0 = \gamma_n = 0$.

The Lorenz group This is the 6-dimensional group $\mathrm{SO}(3,1)$ of 4×4 matrices A such that $AJA^{\mathrm{T}} = J$, where $J = \mathrm{diag}(1,1,1,-1)$ (Carter et al. 1995). It has important applications in special relativity theory. he corresponding *Lorenz algebra* $\mathfrak{so}(3,1)$ consists of all matrices B such that $BJ+JB^{\mathrm{T}} = O$. It is easy to verify that each element of $\mathfrak{so}(3,1)$ can be written in the form

$$B = \begin{bmatrix} 0 & b_1 & b_2 & b_3 \\ -b_1 & 0 & b_4 & b_5 \\ -b_2 & -b_4 & 0 & b_6 \\ b_3 & b_5 & b_6 & 0 \end{bmatrix}, \qquad b_1, b_2, \ldots, b_6 \in \mathbb{R}.$$

Choosing the basis

$$\left\{ \begin{bmatrix} 0&1&0&0 \\ -1&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&1&0 \\ 0&0&0&0 \\ -1&0&0&0 \\ 0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&0 \\ 0&0&1&0 \\ 0&-1&0&0 \\ 0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&1 \\ 0&0&0&0 \\ 0&0&0&0 \\ 1&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 0&1&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{bmatrix} \right\},$$

we obtain the commutator table

$$\begin{array}{lllll}
[V_1,V_2] = -V_3, & [V_1,V_3] = V_2, & [V_1,V_4] = -V_5, & [V_1,V_5] = V_4, & [V_1,V_6] = O, \\
[V_2,V_1] = V_3, & [V_2,V_3] = -V_1, & [V_2,V_4] = -V_6, & [V_2,V_5] = O, & [V_2,V_6] = V_4, \\
[V_3,V_1] = -V_2, & [V_3,V_2] = V_1, & [V_3,V_4] = O, & [V_3,V_5] = -V_6, & [V_3,V_6] = V_5, \\
[V_4,V_1] = V_5, & [V_4,V_2] = V_6, & [V_4,V_3] = O, & [V_4,V_5] = V_1, & [V_4,V_6] = V_2, \\
[V_5,V_1] = -V_4, & [V_5,V_2] = O, & [V_5,V_3] = V_6, & [V_5,V_4] = -V_1, & [V_5,V_6] = V_3, \\
[V_6,V_1] = O, & [V_6,V_2] = -V_4, & [V_6,V_3] = -V_5, & [V_6,V_4] = -V_2, & [V_6,V_5] = -V_3.
\end{array}$$

Thus, out of 180 structure constants, just 24 are nonzero – and they all equal ± 1. After brief claculation, we drive for example the polynomials α_k that yield an order-2 CSK approximant,

$$\begin{aligned}
\alpha_1(t) &= \beta_1 t + \tfrac{1}{2}(\beta_2\beta_3 - \beta_4\beta_5)t^2, \\
\alpha_2(t) &= \beta_2 t - \tfrac{1}{2}(\beta_1\beta_3 + \beta_4\beta_6)t^2, \\
\alpha_3(t) &= \beta_3 t + \tfrac{1}{2}(\beta_1\beta_2 - \beta_5\beta_6)t^2, \\
\alpha_4(t) &= \beta_4 t - \tfrac{1}{2}(\beta_1\beta_5 + \beta_2\beta_6)t^2, \\
\alpha_5(t) &= \beta_5 t + \tfrac{1}{2}(\beta_1\beta_4 - \beta_3\beta_6)t^2, \\
\alpha_6(t) &= \beta_6 t + \tfrac{1}{2}(\beta_3\beta_5 - \beta_2\beta_4)t^2,
\end{aligned}$$

where $B = \sum_{k=1}^{6} \beta_k V_k$.

4 Time symmetry

An approximant $F(tB) \approx \exp(tB)$ is said to be *time symmetric* if $F(tB)F(-tB) = I$, $t \geq 0$. Time symmetric approximants are important for a number of reasons, not least being that they lend themselves to the *Yošida technique,* which allows their order to be increased (Yoshida 1990). The techniques of the last section are not time symmetric. Here we describe their modification, which results in a time-symmetric approximant.

Me mention in passing that it is possible to envisage two distinct techniques to obtain high-order algorithms based on canonical coordinates of the second kind. The first, implicit in the work of the previous section, consists of evaluating the numbers $g_k^{(l)}(0)$ for $l = 1, 2, \ldots, p$, where p is the order of the method. The alternative, the subject matter of the present section, consists in combining a second-order or a fourth-order approximant across a number of steps to obtain a higher-order method.

Given the splitting

$$B = \sum_{l=1}^{s} C_l,$$

it is well known that the *Strang splitting* The approximation

$$F(tB) = \mathrm{e}^{tC_1/2} \cdots \mathrm{e}^{tC_{s-1}/2} \mathrm{e}^{tC_s} \mathrm{e}^{tC_{s-1}/2} \cdots \mathrm{e}^{tC_1/2} \tag{4.1}$$

is of order 2 and time symmetric. Note that, as a consequence of time symmetry, for sufficiently small $t \geq 0$ we can represent $F(tB) = \mathrm{e}^{\mathcal{F}(t)}$ where the matrix function $\mathcal{F}(t)$ is odd. It is precisely this feature that allows the application of the Yošida technique.

The clear reason for (4.1) being time symmetric is that it is palindromic in the alphabet $\{C_1, C_2, \ldots, C_s\}$. This provides a clue how to modify techniques based on canonical coordinates of the second kind so as to render them time symmetric. Given a basis $\{V_1, V_2, \ldots, V_d\}$ of the Lie algebra $\mathfrak{g}$, we approximate e^{tB} by the product

$$\exp[\alpha_1(t)V_1] \cdots \exp[\alpha_{d-1}(t)V_{d-1}] \exp[\alpha_d(t)V_d] \exp[\alpha_{d-1}(t)V_{d-1}] \cdots \exp[\alpha_1(t)V_1], \tag{4.2}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_d$ are *odd* polynomials.

Taking $\alpha_l = \frac{1}{2}\beta_l t$, $l = 1, 2, \ldots, d-1$ and $\alpha_d = \beta_d t$ yeilds the second-order Strang splitting. In the sequel we seek higher-order methods of this kind.

Using the Baker–Campbell–Hausdorff (BCH) formula it is possible to express the product of exponentials at the right hand side of (4.2) as a single exponential (Varadarajan 1984, p. 141). Due to the symmetric arrangement of the exponentials in (4.2), the BCH formula is an expansion in odd powers of t. If this expansion converges, which is always the case for sufficently small t, it makes sense to write the equation

$$tB = 2\sum_{i=1}^{d-1} \alpha_i(t)V_i + \alpha_d(t)V_d + \sum_{k=1}^{\infty} Q^{2k}(\boldsymbol{\alpha}). \tag{4.3}$$

Here we denote by $Q^{2k}(\boldsymbol{\alpha})$ the terms of order $\mathcal{O}(t^{2k+1})$ in the BCH formula applied to (4.3). Moreover, we let $\alpha_i^{2k}(t)$ be the polynomial obtained by truncating the expansion of $\alpha_i(t)$ after the first k terms, and we denote the remainder by $r_i^{2k}(t)$. In other words,

$$\alpha_i(t) = \alpha_i^{2k}(t) + r_i^{2k}(t), \qquad r_i^{2k}(t) = \mathcal{O}(t^{2k+1}), \qquad i = 1, 2, \ldots, d.$$

From (4.3) we deduce

$$2\sum_{i=1}^{d-1} \alpha_i^{2(k)}(t)V_i + \alpha_d^{2(k)}(t)V_d = tB - \sum_{r=1}^{k-1} Q^{2r}(\boldsymbol{\alpha}) + \mathcal{O}(t^{2k+1}).$$

Noting that

$$Q^{2r}(\boldsymbol{\alpha}) = Q^{2r}(\boldsymbol{\alpha}^{2(k-1)} + \boldsymbol{r}^{2(k-1)}) = Q^{2r}(\boldsymbol{\alpha}^{2(k-1)}) + \mathcal{O}(t^{2k+r}),$$

we obtain

$$2\sum_{i=1}^{d-1} \alpha_i^{2(k)}(t)V_i + \alpha_d^{2(k)}(t)V_d = tB - \sum_{r=1}^{k-1} Q^{2r}(\boldsymbol{\alpha}^{2(k-1)}) + \mathcal{O}(t^{2k+1}). \tag{4.4}$$

Dropping the $\mathcal{O}(t^{2k+1})$ terms in (4.4), it is possible to compute $\boldsymbol{\alpha}^{2k}$ from $\boldsymbol{\alpha}^{2(k-1)}$. This gives a procedure to derive a sequence of successively increaing-order approximants of $\exp(tB)$. It is easy to see that the approximants

$$F^{2k}(tB) = \exp(\alpha_1^{2k}(t)V_1) \cdots \exp(\alpha_d^{2k}(t)V_d) \cdots \exp(\alpha_1^{2k}(t)V_1),$$

of $\exp(tB)$ are such that $F^{2k}(tB)F^{2k}(-tB) = I$, hence time symmetry, the reason being the symmetric arrangements of the exponentials in $F^{2k}(tB)$ and the odd-power expansion of the functions α_i^{2k}.

The BCH and symmetric BCH formulae for k-terms have an exceedingly complicated expansion, which can be obtained recursively. In what follows we will make use just of the term $Q^2(\alpha)$, demonstrating how it is possible to compute it explicitely for particular choices of the basis.

In the remainder of this section we consider the implementation of time-symmetric CSK methods. We split B as before and commence by considering the Strang splitting (4.1) except that, to simplify notation, we arrange the terms in reverse ordering,

$$\mathrm{e}^{tC_s/2}\cdots\mathrm{e}^{tC_2/2}\mathrm{e}^{tC_1}\mathrm{e}^{tC_2/2}\cdots\mathrm{e}^{tC_s/2}. \tag{4.5}$$

Lemma 1 *The term Q^2 of the BCH formula applied to (4.5) is*

$$Q^2 = \frac{t^3}{12}\sum_{l=2}^{s}[C_1+\cdots+C_{l-1}+\tfrac{1}{2}C_l,[C_1+\cdots+C_{l-1},C_l]]. \tag{4.6}$$

Proof See the appendix. □

Let us next consider the case $\mathfrak{g} = \mathfrak{so}(n)$, choosing the same sparse basis as in Section 2. Therefore, according to (4.6), we have

$$\begin{aligned} Q^2 = \frac{t^3}{12}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} &[b_{1,2}F_{1,2}+\ldots+b_{i,j-1}F_{i,j-1},[b_{1,2}F_{1,2}+\ldots+b_{i,j-1}F_{i,j-1}, \\ &b_{i,j}F_{i,j}]] + \frac{1}{24}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} b_{i,j}[F_{i,j},[b_{1,2}F_{1,2}+\ldots+b_{i,j-1}F_{i,j-1},b_{i,j}F_{i,j}]]. \end{aligned} \tag{4.7}$$

We compute separately each part of this sum. Exploiting the commutator table of our basis we have

$$\begin{aligned} &[b_{1,2}F_{1,2}+\ldots+b_{i,j-1}F_{i,j-1},b_{i,j}F_{i,j}] \\ &= b_{i,j}\left(-\sum_{s=i+1}^{j-1} b_{i,s}F_{s,j} - \sum_{r=1}^{i-1} b_{r,j}F_{r,i} + \sum_{t=1}^{i-1} b_{t,i}F_{t,j}\right), \end{aligned}$$

and noting that $b_{i,s} = -b_{s,i}$, we deduce that

$$[b_{1,2}F_{1,2}+\ldots+b_{i,j-1}F_{i,j-1},b_{i,j}F_{i,j}] = b_{i,j}\left(\sum_{\substack{t=1\\t\neq i}}^{j-1} b_{t,i}F_{t,j} - \sum_{r=1}^{i-1} b_{r,j}F_{r,i}\right). \tag{4.8}$$

Commuting the right-hand side with $F_{i,j}$ gives

$$\left[F_{i,j},\sum_{\substack{t=1\\t\neq i}}^{j-1} b_{t,i}F_{t,j} - \sum_{r=1}^{i-1} b_{r,j}F_{r,i}\right] = \sum_{\substack{t=1\\t\neq i}}^{j-1} b_{t,i}F_{t,i} + \sum_{r=1}^{i-1} b_{r,j}F_{r,j}. \tag{4.9}$$

Let $\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_n$ be the columns of B and denote

$$\boldsymbol{b}_l^s = \sum_{k=1}^{s-1} b_{k,l}\boldsymbol{e}_k, \qquad k=1,2,\ldots,n.$$

Then (4.8) yields

$$[b_{1,2}F_{1,2}+\ldots+b_{i,j-1}F_{i,j-1},b_{i,j}F_{i,j}]=b_{i,j}(\boldsymbol{b}_i^j\boldsymbol{e}_j^{\mathrm{T}}-\boldsymbol{e}_j\boldsymbol{b}_i^{j\,\mathrm{T}})-b_{i,j}(\boldsymbol{b}_j^i\boldsymbol{e}_i^{\mathrm{T}}-\boldsymbol{e}_i\boldsymbol{b}_j^{i\,\mathrm{T}}),$$

while (4.9) gives

$$\begin{aligned}&[F_{i,j},[b_{1,2}F_{1,2}+\cdots+b_{i,j-1}F_{i,j-1},b_{i,j}F_{i,j}]]\\&=b_{i,j}(\boldsymbol{b}_i^j\boldsymbol{e}_i^{\mathrm{T}}-\boldsymbol{e}_i\boldsymbol{b}_i^{j\,\mathrm{T}})-b_{i,j}(\boldsymbol{b}_j^i\boldsymbol{e}_j^{\mathrm{T}}-\boldsymbol{e}_j\boldsymbol{b}_j^{i\,\mathrm{T}}).\end{aligned}$$

Multiplying the latter by $b_{i,j}$ and summing in i and j we can evaluate (4.7) in n^3 operations. Note that we count separately multiplications and additions, for example, we assume that the cost of Euclidean inner product of two vectors of length n is $2n$ operations.

We now assemble together our results to calculate (4.7). We proceed by splitting the sum $b_{1,2}F_{1,2}+\cdots+b_{i,j-1}F_{i,j-1}$ in three parts, whereby

$$\begin{aligned}&[b_{1,2}F_{1,2}+\cdots+b_{i,j-1}F_{i,j-1},\boldsymbol{b}_i^j\boldsymbol{e}_j^{\mathrm{T}}-\boldsymbol{e}_j\boldsymbol{b}_i^{j\,\mathrm{T}}-(\boldsymbol{b}_j^i\boldsymbol{e}_i^{\mathrm{T}}-\boldsymbol{e}_i\boldsymbol{b}_j^{i\,\mathrm{T}})]\\
&=\sum_{l=1}^{i}\sum_{k=1}^{l-1}[(\boldsymbol{b}_l^l\boldsymbol{e}_l^{\mathrm{T}}-\boldsymbol{e}_l\boldsymbol{b}_l^{l\,\mathrm{T}}),\boldsymbol{b}_i^j\boldsymbol{e}_j^{\mathrm{T}}-\boldsymbol{e}_j\boldsymbol{b}_i^{j\,\mathrm{T}}-(\boldsymbol{b}_j^i\boldsymbol{e}_i^{\mathrm{T}}-\boldsymbol{e}_i\boldsymbol{b}_j^{i\,\mathrm{T}})]\\
&\quad+\sum_{l=i+1}^{j-1}\sum_{k=1}^{i}[(\boldsymbol{b}_l^{i+1}\boldsymbol{e}_l^{\mathrm{T}}-\boldsymbol{e}_l\boldsymbol{b}_l^{i+1\,\mathrm{T}}),\boldsymbol{b}_i^j\boldsymbol{e}_j^{\mathrm{T}}-\boldsymbol{e}_j\boldsymbol{b}_i^{j\,\mathrm{T}}-(\boldsymbol{b}_j^i\boldsymbol{e}_i^{\mathrm{T}}-\boldsymbol{e}_i\boldsymbol{b}_j^{i\,\mathrm{T}})]\\
&\quad+\sum_{l=j}^{m}\sum_{k=1}^{i-1}[(\boldsymbol{b}_l^i\boldsymbol{e}_l^{\mathrm{T}}-\boldsymbol{e}_l\boldsymbol{b}_l^{i\,\mathrm{T}}),\boldsymbol{b}_i^j\boldsymbol{e}_j^{\mathrm{T}}-\boldsymbol{e}_j\boldsymbol{b}_i^{j\,\mathrm{T}}-(\boldsymbol{b}_j^i\boldsymbol{e}_i^{\mathrm{T}}-\boldsymbol{e}_i\boldsymbol{b}_j^{i\,\mathrm{T}})].\end{aligned}$$

Finally,

$$\begin{aligned}&[b_{1,2}F_{1,2}+\cdots+b_{i,j-1}F_{i,j-1},\boldsymbol{b}_i^j\boldsymbol{e}_j^{\mathrm{T}}-\boldsymbol{e}_j\boldsymbol{b}_i^{j\,\mathrm{T}}-(\boldsymbol{b}_j^i\boldsymbol{e}_i^{\mathrm{T}}-\boldsymbol{e}_i\boldsymbol{b}_j^{i\,\mathrm{T}})]\\
&=-\boldsymbol{b}_i^{i\,\mathrm{T}}\boldsymbol{b}_i^jF_{i,j}+\boldsymbol{b}_i^i\boldsymbol{b}_j^{i\,\mathrm{T}}-\boldsymbol{b}_j^i\boldsymbol{b}_i^{i\,\mathrm{T}}+\boldsymbol{b}_j^{i\,\mathrm{T}}\boldsymbol{b}_j^iF_{j,i}-(\boldsymbol{b}_j^i\boldsymbol{b}_i^{j\,\mathrm{T}}-\boldsymbol{b}_i^j\boldsymbol{b}_j^{i\,\mathrm{T}})\\
&\sum_{l=1}^{i-1}b_{l,i}(\boldsymbol{b}_l^l\boldsymbol{e}_j^{\mathrm{T}}-\boldsymbol{e}_j\boldsymbol{b}_l^{l\,\mathrm{T}})-\boldsymbol{b}_l^{l\,\mathrm{T}}\boldsymbol{b}_i^jF_{l,j}-b_{l,j}(\boldsymbol{b}_l^l\boldsymbol{e}_i^{\mathrm{T}}-\boldsymbol{e}_i\boldsymbol{b}_l^{l\,\mathrm{T}})+\boldsymbol{b}_l^{l\,\mathrm{T}}\boldsymbol{b}_j^iF_{l,i}\\
&\sum_{l=i+1}^{j-1}b_{l,i}(\boldsymbol{b}_l^{i+1}\boldsymbol{e}_j^{\mathrm{T}}-\boldsymbol{e}_j\boldsymbol{b}_l^{i+1\,\mathrm{T}})-\boldsymbol{b}_l^{i+1\,\mathrm{T}}\boldsymbol{b}_i^jF_{l,j}+b_{i,l}(\boldsymbol{b}_j^i\boldsymbol{e}_l^{\mathrm{T}}-\boldsymbol{e}_l\boldsymbol{b}_j^{i\,\mathrm{T}})+\boldsymbol{b}_l^{i+1\,\mathrm{T}}\boldsymbol{b}_j^iF_{l,i}\\
&+\sum_{l=j+1}^{n}\boldsymbol{b}_l^{i\,\mathrm{T}}\boldsymbol{b}_j^iF_{l,i}-\boldsymbol{b}_l^{i\,\mathrm{T}}\boldsymbol{b}_i^jF_{l,j}.\end{aligned}$$

We analyse the computational costs of the previous formula, summing over i and j and showing that (4.7) can be computed in $\mathcal{O}(n^3)$ operations. Note that, since $\sum_{l=i+1}^{j-1}b_{i,l}\boldsymbol{e}_l=\boldsymbol{b}_i^i-\boldsymbol{b}_i^j$, we have

$$\boldsymbol{b}_i^i\boldsymbol{b}_j^{i\,\mathrm{T}}-\boldsymbol{b}_j^i\boldsymbol{b}_i^{i\,\mathrm{T}}+\sum_{l=i+1}^{j-1}b_{i,l}(\boldsymbol{b}_j^i\boldsymbol{e}_l^{\mathrm{T}}-\boldsymbol{e}_l\boldsymbol{b}_j^{i\,\mathrm{T}})-(\boldsymbol{b}_j^i\boldsymbol{b}_i^{j\,\mathrm{T}}-\boldsymbol{b}_i^j\boldsymbol{b}_j^{i\,\mathrm{T}})=-2(\boldsymbol{b}_j^i\boldsymbol{b}_i^{j\,\mathrm{T}}-\boldsymbol{b}_i^j\boldsymbol{b}_j^{i\,\mathrm{T}}).$$

It is more convenient to write the previous expression in the form

$$-2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}b_{i,j}(\boldsymbol{b}_j^i\boldsymbol{b}_i^{j\,\mathrm{T}}-\boldsymbol{b}_i^j\boldsymbol{b}_j^{i\,\mathrm{T}})$$

$$= -\sum_{i=1}^{n}\left(\sum_{j=i+1}^{n} 2b_{i,j}\boldsymbol{b}_j^i\right)\boldsymbol{b}_i^{i\,\mathrm{T}} - \boldsymbol{b}_i^i\left(\sum_{j=i+1}^{n} 2b_{i,j}\boldsymbol{b}_j^i\right)^{\mathrm{T}}$$
$$-\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} 2b_{i,j}\left(\boldsymbol{b}_j^i(\boldsymbol{b}_i^j - \boldsymbol{b}_i^i)^{\mathrm{T}} - (\boldsymbol{b}_i^j - \boldsymbol{b}_i^i)\boldsymbol{b}_j^{i\,\mathrm{T}}\right).$$

The first part of this sum is computed in about $\frac{2}{3}n^3$ operations and the second part, exploiting the equality

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} 2b_{i,j}\boldsymbol{b}_j^i(\boldsymbol{b}_i^j - \boldsymbol{b}_i^i)^{\mathrm{T}} = 2\sum_{i=1}^{n-1}\sum_{k=n-1}^{i+2} b_{i,k}\left(\sum_{l=n}^{k+1} b_{i,l}\boldsymbol{b}_l^i\right)\boldsymbol{e}_k^{\mathrm{T}},$$

can also be computed in $\frac{2}{3}n^3$ operations.

Adding terms of the type $\alpha F_{l,j}$ and $\beta F_{l,i}$ leads to

$$-\sum_{l=1}^{i}\boldsymbol{b}_l^{l\,\mathrm{T}}\boldsymbol{b}_i^j F_{l,j} - \sum_{l=i+1}^{j-1}\boldsymbol{b}_l^{i+1\,\mathrm{T}}\boldsymbol{b}_i^j F_{l,j} - \sum_{l=j+1}^{n}\boldsymbol{b}_l^{i\,\mathrm{T}}\boldsymbol{b}_i^j F_{l,j} = -L_i^i\boldsymbol{b}_j^i\boldsymbol{e}_j{}^{\mathrm{T}} - \boldsymbol{e}_j(-L_i^i\boldsymbol{b}_j^j)^{\mathrm{T}},$$

and

$$\sum_{l=1}^{i-1}\boldsymbol{b}_l^{l\,\mathrm{T}}\boldsymbol{b}_j^i F_{l,i} + \sum_{l=i+1}^{j-1}\boldsymbol{b}_l^{i+1\,\mathrm{T}}\boldsymbol{b}_j^i F_{l,i} + \sum_{l=j+1}^{n}\boldsymbol{b}_l^{i\,\mathrm{T}}\boldsymbol{b}_j^i F_{l,i} = L_i^j\boldsymbol{b}_i^j\boldsymbol{e}_i{}^{\mathrm{T}} - \boldsymbol{e}_i(L_i^j\boldsymbol{b}_i^j)^{\mathrm{T}},$$

where the matrix L_i is the lower triangular part of $b_{1,2}F_{1,2} + \ldots + b_{i-1,n}F_{i-1,n}$ and we denote by L_i^s, $s = i, j$, the matrix L_i with zeros along its s-th row. Summing up with respect to i and j, we obtain

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} b_{i,j}(-L_i^i\boldsymbol{b}_j^i)\boldsymbol{e}_i{}^{\mathrm{T}} = \sum_{i=1}^{n-1}(-L_i^i)\left(\sum_{j=i+1}^{n} b_{i,j}\boldsymbol{b}_j^i\right)\boldsymbol{e}_i{}^{\mathrm{T}}$$
$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} b_{i,j}L_i^j\boldsymbol{b}_i^j\boldsymbol{e}_j{}^{\mathrm{T}} = \sum_{j=2}^{n}\left(\sum_{i=1}^{j-1} b_{i,j}\boldsymbol{c}_i\right)\boldsymbol{e}_j{}^{\mathrm{T}}$$

where we have used the notation $\boldsymbol{c}_i := L_i\boldsymbol{b}_i^i$ for $i = 1, \ldots, n-1$. The cost of computing the first sum is $\frac{4}{3}n^3$, while the cost of computing the second is $2n^3$ operations.

Finally the terms

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{l=1}^{i} b_{i,j}b_{l,i}(\boldsymbol{b}_l^l\boldsymbol{e}_j^{\mathrm{T}} - \boldsymbol{e}_j\boldsymbol{b}_l^{l\,\mathrm{T}}) = \sum_{l=1}^{n-1}\boldsymbol{b}_l^l\left(\sum_{j=l+2}^{n} c_{j,l}\boldsymbol{e}_j^{\mathrm{T}}\right) - \left(\sum_{j=l+2}^{n} c_{j,l}\boldsymbol{e}_j\right)\boldsymbol{b}_l^{l\,\mathrm{T}}$$

with $c_{j,l} = \sum_{i=l}^{j-1} b_{i,j}b_{l,i}$,

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{l=1}^{i} b_{i,j}b_{l,j}(\boldsymbol{b}_l\boldsymbol{e}_i^{\mathrm{T}} - \boldsymbol{e}_i\boldsymbol{b}_l^{\mathrm{T}}) = \sum_{l=1}^{n-1}\boldsymbol{b}_l^l\left(\sum_{i=l+1}^{n-1} d_{i,l}\boldsymbol{e}_i^{\mathrm{T}}\right) - \left(\sum_{i=l+1}^{n-1} d_{i,l}\boldsymbol{e}_i\right)\boldsymbol{b}_l^{l\,\mathrm{T}}$$

with $d_{i,l} = \sum_{j=i+1}^{n} b_{i,j}b_{l,j}$, and

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{l=i+1}^{j-1} b_{i,j}b_{l,i}(\boldsymbol{b}_l^{i+1}\boldsymbol{e}_j^{\mathrm{T}} - \boldsymbol{e}_j\boldsymbol{b}_l^{i+1\,\mathrm{T}})$$
$$= \sum_{i=1}^{n-1}\sum_{l=i+1}^{n-1} b_{l,i}\left(\boldsymbol{b}_l^{i+1}(\boldsymbol{b}_i^{l+1} - \boldsymbol{b}_i)^{\mathrm{T}} - (\boldsymbol{b}_i^{l+1} - \boldsymbol{b}_i)\boldsymbol{b}_l^{i+1\,\mathrm{T}}\right);$$

can be computed in about $\frac{2}{3}n^3$, $\frac{2}{3}n^3$ and $\frac{1}{2}n^3$ operations respectively. Collecting the contributions of all the terms in the sum we obtain a total count of $7\frac{1}{2}n^3$ operations.

At the present time it is not clear that this method of computation of Q^2 in the $\mathfrak{so}(n)$ case is optimal form the point of view of complexity theory. We did not try any other ordering of the basis elements and it is not at all certain that different orderings could give better constants in front of the term n^3.

Given that the construction of the (second-order) Strang splitting carries a cost of $4n^3$ operations, the total flop count for constructing a symmetric fourth order SKC approximation of an exponential in $\mathfrak{so}(n)$ by our algorithm is $11\frac{1}{2}n^3$. This is marginally better than obtaining an order-4 approximation by the Yošida technique from three Strang splittings which, as pointed out in (Celledoni & Iserles 1998), requires $12n^3$ flops.

5 Sparse matrices

In a naive formulation, the method of canonical coordinates of the second kind is considerably too expensive for practical computation. This, however, can be alleviated by the use of a sufficiently 'sparse' basis of the underlying Lie algebra $\mathfrak{g}$. As explained in Section 2, choosing a basis so that an overwhelming majority of structure constants vanish renders the algorithm strikingly more effective. It is important to emphasize that this has nothing to do with the structure of the matrix $B \in \mathfrak{g}$, which need not be sparse. Yet, in most practical computations (in particular when n is large) one can expect B to be sparse and structured. Good algorithms should be able to exploit this phenomenon.

In the case of SKC methods we identify two mechanisms that allow us to exploit sparsity. Although this aspect of our methods is still a matter for active investigation, the interim results are substantive enough to warrant publication. For simplicity, we describe the first mechanism just in the case of a tridiagonal $B \in \mathfrak{so}(n)$, hence

$$B = \begin{bmatrix} 0 & \beta_1 & 0 & \cdots & 0 \\ -\beta_1 & 0 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & \beta_{n-1} \\ 0 & \cdots & 0 & -\beta_{n-1} & 0 \end{bmatrix} = \sum_{k=1}^{n-1} \beta_k F_{k,k+1},$$

where the matrices $F_{k,l} = \boldsymbol{e}_k \boldsymbol{e}_l^{\mathrm{T}} - \boldsymbol{e}_l \boldsymbol{e}_k^{\mathrm{T}}$ have been introduced in Section 2. Since

$$b_{k,l} = \begin{cases} \beta_k, & l = k+1, \\ 0, & \text{otherwise}, \end{cases}$$

it is easy to substitute in the general formulae for the order-2 method:

$$g'_{i,j}(0) = \begin{cases} \beta_i, & j = i+1, \\ 0, & \text{otherwise}, \end{cases} \qquad g''_{i,j}(0) = \begin{cases} \beta_{i-1}\beta_i, & j = i-2, \\ 0, & \text{otherwise}, \end{cases} \qquad 1 \le i < j \le n,$$

Arranging the elements of the basis in lexicographic order, we thus obtain the second-order approximant

$$\mathrm{e}^{\beta_{n-1}tF_{n-1,n}}\mathrm{e}^{\beta_{n-2}tF_{n-2,n-1}}\mathrm{e}^{\frac{1}{2}\beta_{n-1}\beta_{n-1}t^2F_{n-2,n}}\cdots\mathrm{e}^{\beta_2tF_{2,3}}\mathrm{e}^{\frac{1}{2}\beta_2\beta_3t^2F_{2,4}}\mathrm{e}^{\beta_1tF_{1,2}}\mathrm{e}^{\frac{1}{2}\beta_1\beta_2t^2F_{1,3}}.$$

In other words, the cost of the approximation is just $\mathcal{O}(n)$ flops.

Similar situation pertains to

$$B = \begin{bmatrix} \gamma_1 & \eta_1 & 0 & \cdots & 0 \\ \mu_1 & \gamma_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \gamma_{n-1} & \eta_{n-1} \\ 0 & \cdots & 0 & \mu_{n-1} & \gamma_n \end{bmatrix} \in \mathfrak{sl}(n).$$

Choosing the same basis and terminology as in Section 2 we can readily ascertain that

$$\begin{aligned} g''_{k,k-2}(0) &= \mu_{k-2}\mu_{k-1} \qquad k = 3,4,\ldots,n \\ g''_{k,k-1}(0) &= -(\gamma_{k-2} - 2\gamma_{k-1} + \gamma_k)\mu_{k-1}, \qquad k = 3,4,\ldots,n, \\ g''_{k,k+1}(0) &= (\gamma_{k-1} - 2\gamma_k + \gamma_{k+1})\eta_k, \qquad k = 2,3,\ldots,n-1, \\ g''_{k,k+2}(0) &= -\eta_k\eta_{k+1}, \qquad k = 1,2,\ldots,n-2, \\ g''_{k,l}(0) &= 0, \qquad |k-l| \geq 3 \end{aligned}$$

and $g''_k(0) = -\eta_k\mu_k$, $k = 1,2,\ldots,n-1$. Thus, a second-order approximant to a tridiagonal $B \in \mathfrak{sl}(n)$ is itself quindiagonal and its computation requires just $\mathcal{O}(n)$ flops.

Higher-order approximants and matrices with greater bandwidth lend themselves to similar treatment, although the savings are less striking. In a sense, the situation is parallel to that of approximating $\exp(tB)$ by a rational approximant, when savings accrue from sparse matrix-inversion methods, except that in our case the result is assured to belong to the right Lie group.

Another observation which is highly pertinent to the approximation of exponentials of sparse matrices has been made in (Iserles 1999). Suppose that B is a banded matrix of bandwidth $s \geq 3$. In general, $F(t) = \exp(tB)$ is a dense matrix. Yet, as is easy to illustrate by computer experiments, $F(t)$ is very near to a banded matrix. Specifically, given $\varepsilon > 0$, there exists $r = r(t,\varepsilon) \geq s$ such that all the elements of $F(t)$ outside a band of width r are less than ε in magnitude. Moreover, tight upper bounds on r can be derived with relative ease. The idea thus is to set to zero all the elements outside bandwidth r. The outcome is a banded approximant to the exponential. Moreover, with an appropriate choice of basis elements, this means that the functions α_i are *set to zero* for elements that possess terms exclusively outside the band. Consequently corresponding exponentials equal identity and need not be included in the product. Thus, the cost scales with the size r of the bandwidth. Similar phenomenon has been already encountered in the context of $\mathfrak{so}(n)$ and $\mathfrak{sl}(n)$, when our choice of basis and order has implied a banded structure of the exponential. The present mechanism is different, even if the net outcome is similar.

6 Numerical experiments

Our numerical experiments are organized as follows. We fist consider a test on random matrices in $\mathfrak{so}(50)$, illustrating the performance of methods based on the use of second kind coordinates techniques for full and sparse matrices. The third and last example is the solution of a third-order ODE using Runge–Kutta/Munthe-Kaas (RK/MK) methods described in (Munthe-Kaas 1997). We use the `Matlab` toolbox `DiffMan` for the integration of ODEs on manifolds, comparing the usual implementation of RK/MK methods, whereby the the exponential is approximated to machine accuracy, with a version of the methods obtained using the time-symmetric fourth-order approximation from Section 4.

All experiments have been performed in `Matlab` and we have computed the error while comparing the results with the built-in function `expm` which calculates the exponential to nearly machine accuracy.

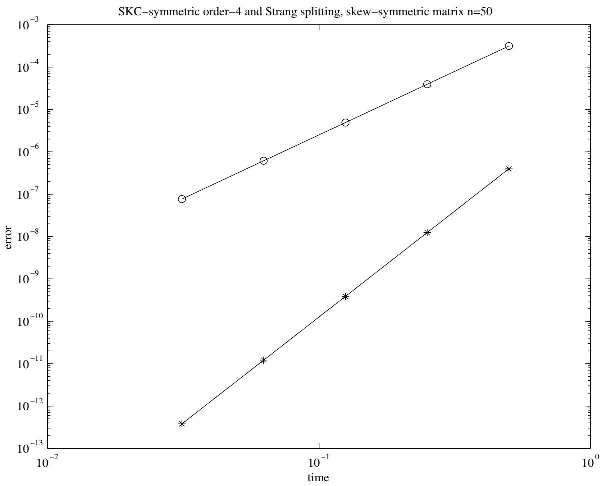

Figure 2: Error versus time in the $\mathfrak{so}(50)$ (full case).

We evaluated the the error computing $\|e^{-tB}F(tB) - I\|_{\mathrm{F}}$ where $F(tB)$ is the SKC approximation of $\exp(tB)$ and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. The matrices have been generated randomly using the `Matlab` function `rand` and scaling the Frobenius norm so that $\|B\|_{\mathrm{F}} = 1$.

We approximate $\exp(tB)$ with a single step of the methods for different values of t, ($t = 1/2^k$ and $k = 1, \ldots, 5$).

In both the first two figures the norm of the error is plotted (along the y-axis) to a logarithmic scale with respect to t. Figure 2 reports the results of our first test, where we have considered a full matrix in $\mathfrak{so}(50)$. In the plots the error norm is indicated with the symbols '$*$' (SKC, time symmetric, order 4) and '$\circ$' (Strang splitting, order 2).

In the next example, illustrated in Figure 3, the same methods have been applied to a sparse matrix in $\mathfrak{so}(50)$, with four non-zero diagonals (i.e., bandwidth 5). In both the examples the methods give the correct order. In the second case, however, the count of flops is drastically reduced. We counted the number of flops using the `Matlab` function `flops`. In the first case the cost for constructing Q^2 amounts to $9.62n^3$ while in the second we counted $0.95n^3$ flops. As it is easy to understand, the described implementation of the methods allows to take advantage immediately of the sparsity structure of the matrix B, working directly on the nonzeros entries of B *à la* Section 4.

The last example is concerned with the use of the techniques described in this paper in substituting the exponentials computed to machine accuracy in the integration methods of (Munthe-Kaas 1997). The experiments have been performed using the `Matlab` toolbox `DiffMan.` We use a RK/MK method of order four.

The example is a problem whose solution is the soliton originating in the *Korteweg–de Vries* (KdV) equation. It is a third-order ODE obtained performing a symmetry reduction on the KdV equation. The resulting ODE can be written as a three-dimensional system,

$$y' = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -9y(2) & 3 & 0 \end{bmatrix} y$$

with $y(0) = [1, 0, -1.5]^{\mathrm{T}}$ and $t \in [0, 5]$. The solution of the ODE $f = y_1(t)$ can be easily derived explicitely and it is $f(t) = \alpha \mathrm{sech}\,(t\beta)$, $\alpha = 1$, $\beta = 1/2\sqrt{(}3)$.

In Figure 4 we plot the analytic solution (solid line) on a grid of 161 points. The dotted line is the numerical solution obtained with the `Matlab` routine `ode45` with absolute and relative tolerance $1.0e-4$. The method produced this solution in 69 steps, and it was implemented with step-size

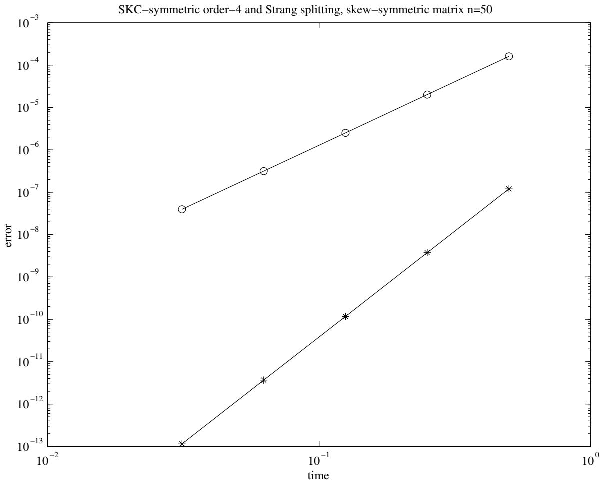

Figure 3: Error versus time in the $\mathfrak{so}(50)$ sparse case.

control procedure. The dashed-dotted line is the numerical solution obtained with the RK/MK method using SKC symmetric tecnhniques for the approximation of the exponential, with fixed step-size h.

In Figure 5 we plot the error (along the y-axis) with respect to the numerical solution obtained with the `Matlab` routine `ode45` to a logarithmic scale, versus the stepsize $h = 1/2^k$ and $k = 1, \ldots, 5$ for the cases of the implementation of RK/MK with the `expm` function of `Matlab` (marked with $+$) and approximating exp to order four with a SKC technique ($\circ$). The line marked with $*$ representes the error of the numerical solution given by the RK/MK method implemented with SKC technique for the approximation of the exponential, measured with respect to the numerical solution obtained by the same method with the use of the exact exponential (`expm` routine of `Matlab`).

It is interesting to note in this case that substituting the exact exponential with suitable fourth-order approximant does not lead to a significant deterioration in the quality of the RK/MK method and the overall error does not change much. Note that in the present case the primary variable is a vector, rather than a matrix. In general, if the underlying ODE can be written in a vector form, i.e. as an action of a Lie group on $\mathbb{R}^n$, we need to approximate $\exp(tB)\boldsymbol{v}$, where $\boldsymbol{v} \in \mathbb{R}^n$, rather than the matrix $\exp(tB)$. This leads to obvious savings in the SKC techniques, similarly, say, to the approach of rational functions. In particular, the cost of composing exponentials is $\mathcal{O}(n^2)$, rather than $\mathcal{O}(n^3)$, operations.

Acknowledgments

The authors are grateful to Brynjulf Owren for many fruitful discussions, to Per Christian Moan for bringing the reference (Wei & Norman 1963) to their attention, to the Numerical Analysis group of DAMTP Cambridge, and to the Geometric Integration members during the fall semester 1998 at MSRI Berkeley. Research at MSRI is supported in part by NSF grant DMS-9701755.

References

Carter, R., Segal, G. & Macdonald, I. (1995), *Lectures on Lie Groups and Lie Algebras*, LMS Student Texts, Cambridge University Press, Cambridge.

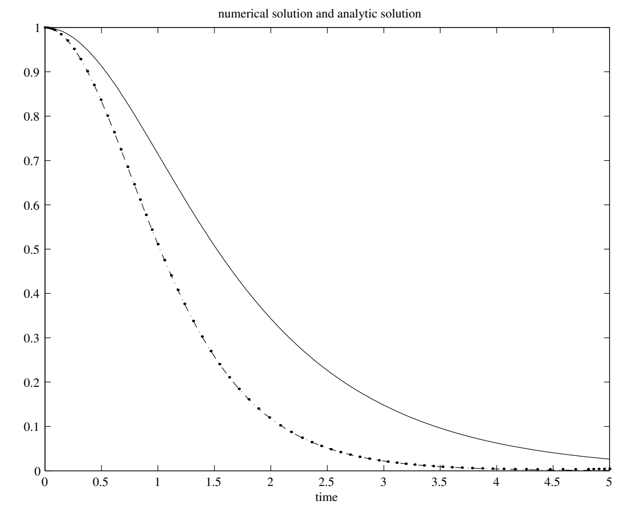

Figure 4: The soliton originating in the KdV equation.

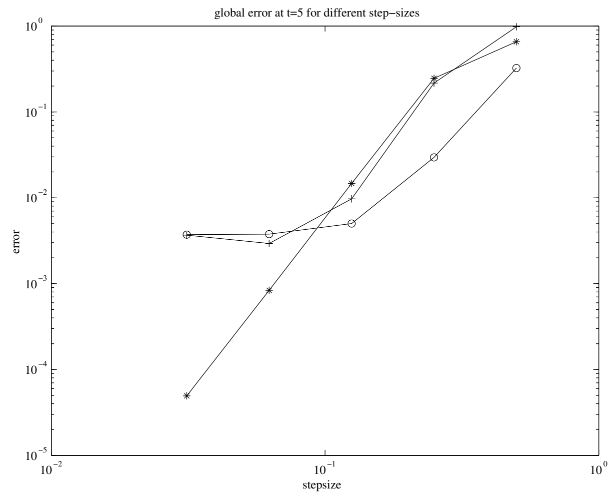

Figure 5: RK/MK: global error at $t = 5$ with `expm` and SKC

Casas, F. (1996), 'Fer's factorization as a symplectic integrator', *Numer. Math.* **74**(3), 283–303.

Celledoni, E. & Iserles, A. (1998), Approximating the matrix exponential from a Lie algebra to a Lie group, Technical Report 1998/NA03, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England.

Chacon, R. & Fomenko, A. (1991), 'Recursion formulas for the Lie integral', *Advances in Math.* **88**, 200–257.

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

Engø, K. (1998), On the construction of geometric integrators in the RKMK class, Technical Report 158, Department of Informatics, University of Bergen, Norway.

Hochbruck, M., Lubich, C. & Selhofer, H. (1998), 'Exponential integrators for large systems of differential equations', *SIAM J. Sci. Comput.* **19**, 1552–1574.

Humphreys, J. E. (1972), *Introduction to Lie algebras and Representation Theory*, First edn, Springer.

Iserles, A. (1999), How large is the exponential of a banded matrix?, Technical Report DAMTP 1999/1, University of Cambridge.

Iserles, A. & Nørsett, S. P. (1997), Linear ODEs in Lie groups, Technical Report 1997/NA9, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England.

Iserles, A. & Nørsett, S. P. (1999), 'On the solution of linear differential equations in Lie groups', *Phil. Trans. Royal Soc. A.* To appear.

Munthe-Kaas, H. (1997), High order Runge-Kutta methods on manifolds, Technical Report NA14, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England.

Owren, B. & Marthinsen, A. (1997), Runge-Kutta methods adapted to manifolds and based on rigid frames, Technical Report Numerics No. 1/1997, Department of Mathematical Sciences, The Norwegian University of Science and Technology. To appear in BIT.

Owren, B. & Marthinsen, A. (1998), Integration methods based on canonical coordinates of the second kind, In preparation.

Varadarajan, V. S. (1984), *Lie Groups, Lie Algebras, and their Representation*, GTM 102, Springer-Verlag.

Wei, J. & Norman, E. (1963), 'On global representations of the solutions of linear differential equations as a product of exponentials', *Advances in Mathematics*.

Yoshida, H. (1990), 'Construction of higher order symplectic integrators', *Physics Letters A* **150**, 262–268.

Zanna, A. (1997), Collocation and relaxed collocation for the Fer and the Magnus expansions, Technical Report 1997/NA17, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England.

A Appendix

For completeness, we present a proof of Lemma 1. Note that a comprehensive treatment of this subject matter, inclusive of the non-symmetric case, has been presented in a different context by Chacon & Fomenko (1991). For our purposes, however, it is sufficient to derive the first term of the expansion.

Lemma 1 *The leading error term in the Strang splitting is*

$$\frac{1}{12}\sum_{l=2}^{s}[C_1+\cdots+C_{l-1}+\tfrac{1}{2}C_l,[C_1+\cdots+C_{l-1},C_l]]. \tag{4.6}$$

Proof Letting

$$\begin{aligned}
F_1(t) &= \mathrm{e}^{tC_1},\\
F_l(t) &= \mathrm{e}^{tC_l/2}F_{l-1}\mathrm{e}^{tC_l/2}, \qquad l=2,3,\ldots,s,
\end{aligned}$$

we can verify at once that F_s is precisely the Strang splitting. We assume that

$$F_l(t)=\exp[t(C_1+\cdots+C_l)+\tfrac{1}{12}Q_l t^3+\mathcal{O}(t^4)], \qquad l=1,2,\ldots,s.$$

We use the BCH formula:

$$\begin{aligned}
F_l(t) &= \exp(\tfrac{1}{2}tC_l)\exp[t(C_1+\cdots+C_{l-1})+\tfrac{1}{12}Q_{l-1}t^3+\mathcal{O}(t^4)]\exp(\tfrac{1}{2}tC_l)\\
&= \exp\{t(C_1+\cdots+C_{l-1}+\tfrac{1}{2}C_l)+\tfrac{1}{4}t^2[C_l,C_1+\cdots+C_{l-1}]\\
&\quad +\tfrac{1}{24}t^3[\tfrac{1}{2}C_l-(C_1+\cdots+C_{l-1}),[C_l,C_1+\cdots+C_{l-1}]]\\
&\quad +\tfrac{1}{12}t^3Q_{l-1}+\mathcal{O}(t^4)\}\exp(\tfrac{1}{2}tC_l)\\
&= \exp\{t(C_1+\cdots+C_l)+\tfrac{1}{4}t^2[C_l,C_1+\cdots+C_{l-1}]\\
&\quad +\tfrac{1}{24}t^3[\tfrac{1}{2}C_l-(C_1+\cdots+C_{l-1}),[C_l,C_1+\cdots+C_{l-1}]]\\
&\quad +\tfrac{1}{4}t^2[C_1+\cdots+C_{l-1}+\tfrac{1}{2}C_l,C_l]+\tfrac{1}{16}t^3[[C_l,C_1+\cdots+C_{l-1}],C_l]\\
&\quad +\tfrac{1}{24}t^3[C_1+\cdots+C_{l-1},[C_1+\cdots+C_{l-1}+\tfrac{1}{2}C_l,C_l]]+\tfrac{1}{12}t^3Q_{l-1}+\mathcal{O}(t^4)\}.
\end{aligned}$$

However,

$$[C_l,C_1+\cdots+C_{l-1}]+[C_1+\cdots+C_{l-1}+\tfrac{1}{2}C_l,C_l]=O,$$

thereby annihilating the t^2 term, and

$$\begin{aligned}
&\tfrac{1}{24}[\tfrac{1}{2}C_l-(C_1+\cdots+C_{l-1}),[C_l,C_1+\cdots+C_{l-1}]]+\tfrac{1}{16}[[C_l,C_1+\cdots+C_{l-1}],C_l]\\
&\quad +\tfrac{1}{24}[C_1+\cdots+C_{l-1},[C_1+\cdots+C_{l-1}+\tfrac{1}{2}C_l,C_l]]\\
&= \tfrac{1}{12}[C_1+\cdots+C_{l-1}+\tfrac{1}{2}C_l,[C_1+\cdots+C_{l-1},C_l]].
\end{aligned}$$

Therefore

$$Q_l=[C_1+\cdots+C_{l-1}+\tfrac{1}{2}C_l,[C_1+\cdots+C_{l-1},C_l]]+Q_{l-1}.$$

Since $Q_1=O$, the expression (4.6) follows by summing the above formula for $l=2,3,\ldots,s$. □