EFFICIENT COMPUTATION OF THE MATRIX EXPONENTIAL BY GENERALIZED POLAR DECOMPOSITIONS

ARIEH ISERLES * AND ANTONELLA ZANNA †

Abstract. In this paper we explore the computation of the matrix exponential in a manner that is consistent with Lie-group structure. Our point of departure is the method of generalized polar decompositions, which we modify and combine with similarity transformations that bring the underlying matrix to a form more amenable to efficient computation. We develop techniques valid for a range of Lie-groups: the orthogonal group, the symplectic group, Lorenz, isotropy and scaling groups. However, the GPD approach is equally promising in a more general context: even when Lie-group structure is not at issue, our algorithm is more efficient in many settings than classical methods for the computation of the matrix exponential.

Key words. Matrix exponential, Lie group, Lie algebra, generalized polar decomposition.

AMS subject classifications. 65F30

1. Introduction. The approximation of the matrix exponential is among the oldest and most extensively researched problems in numerical mathematics. Yet, nineteen dubious ways (Moler & van Loan 1978) and many efficient algorithms (cf., for example, (Hochbruck, Lubich & Selhofer 1998)) later, the problem is far from being satisfactorily solved and many challenges remain. This is true in particular when we wish to approximate an exponential of a matrix Z, say, which resides in a *Lie algebra.* This is a central problem in *geometric integration,* which arises once we wish to discretise systems of differential equations evolving in *Lie groups* (smooth manifolds with group structure) and in *homogeneous manifolds* (smooth manifolds which are subjected to transitive group action).

While referring the reader to (Iserles, Munthe-Kaas, Nørsett & Zanna 2000) for a substantive survey of Lie-group methods and their applications, and to Section 2 for formal definitions, it is important to mention informally a number of salient features of such methods, since they motivate much of the work of the present paper.

- The tangent space $\mathrm{T}_x G$, where G is a Lie group and $x \in G$, is $\{Zx : Z \in \mathfrak{g}\}$, where $\mathfrak{g} = \mathrm{T}_I G$ and I is the identity of G. Therefore, once we know $\mathfrak{g}$, we can describe all vector fields (hence, all differential equations) on G.
- The linear space $\mathfrak{g}$ is a Lie algebra: it is closed under an antisymmetric binary operation of *commutation.*
- The *exponential map* takes the Lie algebra to 'its' Lie group, $\exp \mathfrak{g} \subseteq G$.
- Most finite-dimensional Lie groups in practical applications are comprised of matrices. Familiar examples are the general linear group $\mathrm{GL}(\mathbb{R}, n)$ ($n \times n$ nonsingular real matrices), the special linear group $\mathrm{SL}(\mathbb{R}, n)$ ($n \times n$ real matrices with unit determinant) and the orthogonal group $\mathrm{O}(\mathbb{R}, n)$ ($n \times n$ real orthogonal matrices).
- All finite-dimensional Lie algebras are isomorphic to Lie algebras of matrices. In particular, the Lie algebras corresponding to the three Lie groups above are $\mathfrak{gl}(\mathbb{R}, n)$ (the $n \times n$ real matrices), $\mathfrak{sl}(\mathbb{R}, n)$ ($n \times n$ real matrices with zero trace) and $\mathfrak{so}(\mathbb{R}, n)$ ($n \times n$ real skew-symmetric matrices), respectively.

*Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England, email A.Iserles@damtp.cam.ac.uk.

†Department of Informatics, University of Bergen, Høyteknologisenteret, N-5020 Bergen, Norway, email Antonella.Zanna@ii.uib.no.

- If G is a matrix group (hence, $\mathfrak{g}$ is a matrix algebra) the operations of commutation and exponentiation are the familiar matricial commutator and exponent, respectively.

Therefore, once a differential equation evolves in a matrix Lie group, it can be always written in the form

$$y' = F(t,y)y, \quad t \geq 0, \qquad y(0) \in G, \tag{1.1}$$

where $F : \mathbb{R}_+ \times G \to \mathfrak{g}$. Moreover, its solution can be represented (subject to the usual caveats of convergence) in the form $y(t) = \exp(\Omega(t))y(0)$, where Ω evolves in the Lie algebra $\mathfrak{g}$. It is possible to replace (1.1) by an equation for Ω, which evolves in $\mathfrak{g}$ (Iserles et al. 2000), and there are important benefits in solving the latter, returning to the Lie group in every time step by means of the exponential map. The main advantage is that $\mathfrak{g}$ is a linear space and, as long as we discretise equations therein employing exclusively linear-space operations and commutators, we can be assured that the numerical solution stays in $\mathfrak{g}$. Thus, once exponentiated, we obtain a numerical solution that evolves in the Lie group: this is important in the many instances when the preservation of Lie-group structure is important and in variance with most numerical methods applied directly in G (Iserles et al. 2000).

The above argument is at the heart of many Lie-group methods (Runge–Kutta–Munthe-Kaas schemes, Magnus expansions). Other methods, based on different premises (e.g. Crouch–Grossman methods, Fer expansions and methods based upon canonical coordinates of the second kind) also require the computation (or approximation) of the matrix exponential. However, standard methods for the approximation of the matrix exponential, e.g. Padé approximations and Krylov subspace techniques, are not guaranteed to map elements from $\mathfrak{g}$ to G: thus, having gone to a great length to respect Lie-algebraic structure, we might well loose the fruits of this endeavour while computing the exponential! On the positive side, diagonal Padé approximations map some Lie algebras ("quadratic" algebras: $\mathfrak{so}(\mathbb{R},n)$, the symplectic algebra, the Lorenz algebra) to the underlying group. However, it is possible to show that the only analytic function f, that maps $\mathfrak{sl}(\mathbb{R},n)$ into $\mathrm{SL}(\mathbb{R},n)$ consistently with the exponential function (i.e., $f(z) = 1 + z + \mathcal{O}(z^2)$) is the exponential itself (Kang & Shang 1995). Also other classical methods for the approximation of the exponential fail in that case, and this motivates the development of new breeds of approximation algorithms.

Early inroads into the approximation of the exponential in a Lie-algebraic setting have been made in (Celledoni & Iserles 2000), using the splitting approach,

$$\mathrm{e}^{tZ} \approx \mathrm{e}^{tV_1}\mathrm{e}^{tV_2}\cdots\mathrm{e}^{tV_m},$$

where each V_k resides in $\mathfrak{g}$ and the computation of its exponential is easy. The latter is true when the V_ks are of low rank and this, indeed, was the approach introduced in (Celledoni & Iserles 2000).

Suppose that $\dim\mathfrak{g} = s$ and let $\boldsymbol{X} = \{X_1, X_2, \ldots, X_s\}$ be a basis of $\mathfrak{g}$. In that case it is possible to represent $\exp(tZ)$ for $Z \in \mathfrak{g}$ and sufficiently small $|t|$ in *canonical coordinates of the second kind,*

$$\mathrm{e}^{tZ} = \mathrm{e}^{g_1(t)X_1}\mathrm{e}^{g_2(t)X_2}\cdots\mathrm{e}^{g_s(t)X_s},$$

where the scalar functions g_k are analytic at the origin. Although the g_ks are implicitly defined, it is possible to approximate their truncated Taylor expansion, an approach

adopted in (Celledoni & Iserles 2001). A naive procedure of this kind might be excessively expensive, but the cost can be reduced by several orders of magnitude by a clever choice of the basis $\boldsymbol{X}$, exploiting the Lie-algebraic structure.

The work underlying the approach of the present paper, *generalized polar decompositions (GPD),* has been introduced in (Munthe-Kaas, Quispel & Zanna 2001) and further elaborated in (Zanna 2000, Zanna & Munthe-Kaas 2002). In Section 2 we present a brief review of such methods. It suffices to state here that, while building upon former work in this area, they establish a general framework which leads to robust and affordable algorithms. Having said this, such algorithms can be fairly expensive when the required order is high, in particular when they need to be computed, perhaps repeatedly, in each time step. The purpose of this paper is to bring together generalized polar decompositions with techniques from numerical linear algebra, thereby leading to more efficient and cheaper algorithms.

At a conceptual level, we are attempting to marry two types of structures, which are often incompatible. For example, a viable approach to compute $\exp tZ$ for $Z \in \mathfrak{gl}(\mathbb{R}, n)$ is to represent $Z = VHV$, where V is a product of Householder reflections and H is upper Hessenberg. Since this is a similarity transformation, it is true that $\mathrm{e}^{tZ} = V\mathrm{e}^{tH}V$ and we need to compute an exponential of an upper-Hessenberg matrix. As we show in the sequel, this can be done very efficiently indeed by a modification of the GPD technique. Unfortunately, this approach cannot be extended to other Lie algebras. Thus, suppose that Z resides in the *symplectic algebra*

$$\mathfrak{sp}(n) = \{Y \in \mathfrak{gl}(\mathbb{R}, 2n) \,:\, YJ + JY^{\top} = O\}, \qquad \text{where} \qquad J = \begin{bmatrix} O & I \\ -I & O \end{bmatrix}.$$

In that case, in general, $H \notin \mathfrak{sp}(n)$: a Hessenberg form and symplecticity are incompatible! This is an illustration of a more general state of affairs, when numerical-algebraic and Lie-algebraic structures clash. In this paper we present numerical-algebraic structures which are compatible with a long list of matrix Lie algebras that occur in applications. Moreover, in each case we need to modify and fine-tune the GPD algorithm to reduce its cost and improve its efficiency.

This is the point to mention that the GPD approach, combined with an upper-Hessenberg form and the "peel-up" technique, result in an algorithm that compares favourably, in terms of both cost and accuracy, with classical methods to compute the exponential of a matrix. Thus, a procedure motivated by retention of specialised differential-geometric structure, and based on mathematics which might be unfamiliar to many numerical analysts, is very valuable also in a general context, where Lie-group structure is not at issue.

The plan of this paper is as follows. In Section 2 we consider in greater detail Lie groups and Lie algebras, introducing requisite theory and notation. This is followed by a brief review of generalized polar decompositions by means of involutory automorphisms. In Section 3 we debate the computation, using GPD, of exponentials of tridiagonal matrices. We introduce a new approach (the "peel-up" algorithm) which renders the GPD method substantially more efficient in this setting. The theme of Section 4 is how to bring matrices to an upper Hessenberg form, or alternative forms that lend themselves to our approach, by means of similarity transformations. Thus, for example, symplectic matrices are converted into a so-called butterfly form. We discuss the implementation and cost of the "peel-up" technique in all these settings. Section 5 is devoted to a divide-and-conquer strategy, which, approximating the exponential of a matrix by computations in lower-dimensional spaces, which can be

performed in unison, lends itself to implementation in parallel architectures. This strategy is fully compatible with GPD and the retention of Lie-group structure and it again displays the merits of the "peel-up" approach. Finally, in Section 6 we discuss the calculation of the exponential by GPD and the "peel-up" technique for a range of more 'exotic' Lie groups: the Lorenz group, the isotropy group and the scaling group. The paper concludes with an appendix, to which we have relegated some of the more technical calculations.

The issue of stability and conditioning is outside the scope of this paper. Although much of the underlying framework, based upon similarity transformations by orthogonal matrices, is consistent with good conditioning, stability might become an issue. We plan to return to this subject-area in a subsequent paper, where we explore in detail the stability of the GPD technique.

2. Background theory. The natural setting of generalized polar decompositions is Lie-group and Lie-algebra theory, therefore it is convenient to present the background theory in the language of differential geometry. To distinguish between group and algebra elements, it is usual in differential geometry to denote Lie-group elements with lower-case letters and Lie-algebra elements with upper-case letters, whether they represent matrices, vectors or scalars (Helgason 1978). An arbitrary Lie group will be denoted by G and the corresponding Lie algebra by $\mathfrak{g}$. Subspaces of $\mathfrak{g}$ are also usually denoted by Gothic letters. We adopt this convention throughout this subsection. Later on, when most of the computations take place at the algebra level, we will revert to a language that is more familiar to the numerical analysis community and matrices (except when we want to emphasise the Lie-group context) will be denoted as usual with capital letters.

Let $G \subseteq \mathrm{GL}(\mathbb{R}, n)$ be a matrix Lie group with Lie algebra $\mathfrak{g}$. Given an involutive automorphism σ of G, i.e. a one-to-one map $G \to G$ such that

$$\begin{aligned} &\sigma(x \cdot y) = \sigma(x) \cdot \sigma(y) \qquad \forall x, y \in G, \\ &\sigma(\sigma(x)) = x \quad \forall x \in G, \qquad \sigma \neq \mathrm{id}, \end{aligned}$$

it is possible to show that, for t sufficiently small, every element $z = \exp(tZ) \in G$, $Z \in \mathfrak{g}$, can be factorised in the form

$$z = xy \tag{2.1}$$

where $\sigma(y) = y$ and $\sigma(x) = x^{-1}$ (Lawson 1994, Munthe-Kaas et al. 2001). The decomposition (2.1) is called the *generalized polar decomposition* of z, in analogy with the case of real matrices with the special choice $\sigma(z) = z^{-\top}$, when it reduces to the familiar polar decomposition.

The automorphism σ induces in a natural manner an involutive automorphism $\mathrm{d}\sigma$ on the Lie algebra $\mathfrak{g}$,

$$\mathrm{d}\sigma(Z) = \frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0} \sigma(\exp(tZ)), \tag{2.2}$$

which defines a splitting of $\mathfrak{g}$ into the direct sum of two linear spaces,

$$\mathfrak{g} = \mathfrak{p} \oplus \mathfrak{k}, \tag{2.3}$$

where $\mathfrak{k} = \{Z \in \mathfrak{g} : \mathrm{d}\sigma(Z) = Z\}$ is a subalgebra of $\mathfrak{g}$, while the set $\mathfrak{p} = \{Z \in \mathfrak{g} : \mathrm{d}\sigma(Z) = -Z\}$ has the structure of a Lie triple system, a linear space closed under the double commutator,

$$A, B, C \in \mathfrak{p} \qquad \Longrightarrow \qquad [A, [B, C]] \in \mathfrak{p},$$

where the bracket $[A, B] = AB - BA$ is the standard matrix commutator.

To show that (2.3) is true, denote by $\Pi_{\mathfrak{p}} : \mathfrak{g} \to \mathfrak{p}$ the canonical projection of $\mathfrak{g}$ onto the subspace $\mathfrak{p}$ and by $\Pi_{\mathfrak{k}} : \mathfrak{g} \to \mathfrak{k}$ its projection onto $\mathfrak{k}$. Set

$$P = \Pi_{\mathfrak{p}}(Z), \qquad K = \Pi_{\mathfrak{k}}(Z).$$

It is easily verified by direct computation that every element Z can be written in a unique manner as $Z = P + K$, where

$$P = \Pi_{\mathfrak{p}}(Z) = \frac{1}{2}(Z - \mathrm{d}\sigma(Z)), \qquad K = \Pi_{\mathfrak{k}}(Z) = \frac{1}{2}(Z + \mathrm{d}\sigma(Z)).$$

To keep our presentation relevant to the subject matter of this paper, we refer the reader to (Munthe-Kaas et al. 2001, Zanna 2000) and references therein for a more extensive treatment of such decompositions. However, it is of fundamental importance to note that the sets $\mathfrak{k}$ and $\mathfrak{p}$ possess the following properties:

$$[\mathfrak{k}, \mathfrak{k}] \subseteq \mathfrak{k}, \qquad [\mathfrak{k}, \mathfrak{p}], [\mathfrak{p}, \mathfrak{k}] \subseteq \mathfrak{p}, \qquad [\mathfrak{p}, \mathfrak{p}] \subseteq \mathfrak{k}, \tag{2.4}$$

meaning that for all $K_1, K_2 \in \mathfrak{k}$ and $P_1, P_2 \in \mathfrak{p}$, it is true that $[K_1, K_2] \in \mathfrak{k}$, $[K_1, P_1], [P_2, K_2] \in \mathfrak{p}$ and $[P_1, P_2] \in \mathfrak{k}$.

How does the splitting of $Z = P + K$ relate to the factorization (2.1)? It is possible to show that, for t sufficiently small, the factors x and y in (2.1) are of the form $x = \exp(X(t))$ and $y = \exp(Y(t))$, where $X(t) \in \mathfrak{p}$ and $Y(t) \in \mathfrak{k}$, for all $t \in [0, t_0]$. Moreover they can be expanded in series

$$X(t) = \sum_{i=1}^{\infty} X_i t^i, \qquad Y(t) = \sum_{i=1}^{\infty} Y_i t^i,$$

where the coefficients X_i and Y_i can be calculated by means of explicit recurrence relations from the matrices P and K (Zanna 2000). The first terms in the expansions of $X(t)$ and $Y(t)$ are

$$\begin{aligned} X = {} & Pt - \tfrac{1}{2}[P, K]t^2 - \tfrac{1}{6}[K, [P, K]]t^3 \\ & + \Big(\tfrac{1}{24}[P, [P, [P, K]]] - \tfrac{1}{24}[K, [K, [P, K]]]\Big)t^4 \\ & + \Big(\tfrac{7}{360}[K, [P, [P, [P, K]]]] - \tfrac{1}{120}[K, [K, [K, [P, K]]]] \\ & - \tfrac{1}{180}[[P, K], [P, [P, K]]]\Big)t^5 + \mathcal{O}\big(t^6\big)\,, \end{aligned} \tag{2.5}$$

$$\begin{aligned} Y = {} & Kt - \tfrac{1}{12}[P, [P, K]]t^3 + \Big(\tfrac{1}{120}[P, [P, [P, [P, K]]]] \\ & + \tfrac{1}{720}[K, [K, [P, [P, K]]]] - \tfrac{1}{240}[[P, K], [K, [P, K]]]\Big)t^5 + \mathcal{O}\big(t^7\big)\,. \end{aligned}$$

Since $X(t)$ and $Y(t)$ and their truncations live in $\mathfrak{p}$ and $\mathfrak{k}$ respectively, it is clearly desirable to choose automorphisms σ such that exponentials of elements in $\mathfrak{p}$ (and eventually $\mathfrak{k}$) and repeated commutators of P and K are easy to compute.

Assume next that $\sigma_1, \sigma_2, \ldots, \sigma_m$ is sequence of involutive automorphisms on G that satisfies the above conditions. Then, taking $\sigma \equiv \sigma_1$, we partition $\mathfrak{g} = \mathfrak{p}_1 \oplus \mathfrak{k}_1$, and approximate

$$\exp(tZ) \approx \exp(X^{[1]}(t)) \exp(Y^{[1]}(t)), \tag{2.6}$$

where $X^{[1]}$ and $Y^{[1]}$ are truncations of (2.5) of suitable order.

By the same token, $\mathfrak{k}_1$ is partitioned as $\mathfrak{p}_2 \oplus \mathfrak{k}_2$ by means of the automorphism σ_2, and

$$\exp(Y^{[1]}(t)) \approx \exp(X^{[2]}(t)) \exp(Y^{[2]}(t)),$$

where again $X^{[2]}$ and $Y^{[2]}$ are truncations of (2.5) of suitable order.

The procedure is iterated for m steps, say, so that $\mathfrak{k}_m$ is of low dimension and therefore exponentials of its elements are easy to compute exactly.

This algorithm approximates $\exp(tZ)$ to a given order of accuracy. In this circumstances, (2.6) will read

$$\exp(tZ) \approx F(t,Z) = \exp(X^{[1]}(t)) \cdots \exp(X^{[m]}(t)) \exp(Y^{[m]}(t)) \tag{2.7}$$

and it corresponds to the algebra direct-sum decomposition

$$\mathfrak{g} = \mathfrak{p}_1 \oplus \cdots \oplus \mathfrak{p}_m \oplus \mathfrak{k}_m. \tag{2.8}$$

In some circumstances, it might be more convenient to use a mirrored form of (2.7),

$$\exp(tZ) \approx \exp(\tilde{Y}^{[m]}(t)) \exp(\tilde{X}^{[m]}(t)) \cdots \exp(\tilde{X}^{[1]}(t)). \tag{2.9}$$

Clearly, the functions $\tilde{Y}(t)$ and $\tilde{X}(t)$ and their truncations are related to $Y(t)$ and $X(t)$. Indeed, it is easily verified that

$$\tilde{Y}(t) = Y(t), \qquad \tilde{X}(t) = -X(-t)$$

(Munthe-Kaas et al. 2001).

2.1. On the choice of automorphisms. ¿From this point onward, we are mostly interested in the algebraic setting, therefore we revert our notation to the more familiar in numerical analysis. Matrices (otherwise specified) will be denoted by capital letters, vectors by boldface letters, et cetera.

To obtain the algebra splitting (2.8), Zanna & Munthe-Kaas (2002) suggested to use automorphisms of the type

$$\sigma(z) = \mathrm{Ad}_S Z = SzS, \qquad z \in G, \tag{2.10}$$

(inner automorphisms) where $S \in \mathrm{O}(n) \cap G$ is a suitable involutory matrix (i.e. $S^2 = I$) such that $SzS \in G$. Note that, at the algebra level, (2.2) implies that

$$\mathrm{d}\sigma(Z) = \mathrm{Ad}_S Z = SZS,$$

in other words, $\mathrm{d}\sigma$ and σ are essentially of the same form.

Consider next an inner automorphism Ad_S, where

$$S = \mathrm{diag}(-1)^{\mathbf{s}} = \mathrm{diag}[(-1)^{s_1}, (-1)^{s_2}, \ldots, (-1)^{s_n}] \qquad s_i \in \{0,1\}. \tag{2.11}$$

Given an arbitrary matrix Z, one has

$$(SZS)_{i,j} = (-1)^{s_i+s_j} Z_{i,j}$$

consequently,

$$P_{i,j} = \tfrac{1}{2}[1-(-1)^{s_i+s_j}]Z_{i,j}, \qquad K_{i,j} = \tfrac{1}{2}[1+(-1)^{s_i+s_j}]Z_{i,j}.$$

Hence, choosing appropriately the vector $\mathbf{s}$, it is possible to dispatch selected rows and columns of Z to the different subspaces.

For instance, choosing $s_i = 1, i \neq k$ and $s_k = -1$, we obtain subspaces $\mathfrak{p}$ and $\mathfrak{k}$ with the sparsity structure

$$\mathfrak{p} \ni \begin{bmatrix} 0 & \cdots & 0 & \times & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \times & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \times & 0 & 0 & \cdots & 0 \\ \times & \times & \times & 0 & \times & \times & \times & \times \\ 0 & \cdots & 0 & \times & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \times & 0 & 0 & & 0 \\ \vdots & & \vdots & \times & \vdots & & & \vdots \\ 0 & \cdots & 0 & \times & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathfrak{k} \ni \begin{bmatrix} \times & \cdots & \times & 0 & \times & \times & \cdots & \times \\ \vdots & & \vdots & 0 & \vdots & \vdots & & \vdots \\ \times & \cdots & \times & 0 & \times & \times & \cdots & \times \\ 0 & 0 & 0 & \times & 0 & 0 & 0 & 0 \\ \times & \cdots & \times & 0 & \times & \times & \cdots & \times \\ \times & \cdots & \times & 0 & \times & \times & \cdots & \times \\ \vdots & & \vdots & 0 & \vdots & \vdots & & \vdots \\ \times & \cdots & \times & 0 & \times & \times & \cdots & \times \end{bmatrix}$$

respectively.

3. Tridiagonal matrices. Let us assume that Z is a tridiagonal matrix,

$$Z = \begin{bmatrix} \alpha_1 & \gamma_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \gamma_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix}. \tag{3.1}$$

and denote by $\mathbf{e}_i$ the ith unit vector in $\mathbb{R}^n$.

DEFINITION 3.1. *We say that the sequence of automorphisms*

$$\mathrm{Ad}_{S_i}, \quad S_i = \mathrm{diag}(-1)^{\mathbf{s}_i}, \quad i = 1, \ldots, n-1, \tag{3.2}$$

constitutes a peel-down approach *if* $\mathbf{s}_1 = \mathbf{e}_1 = [1, 0, 0, \ldots, 0]^\top$, $\mathbf{s}_2 = \mathbf{e}_2 = [0, 1, 0, \ldots, 0]^\top$,..., $\mathbf{s}_{n-1} = \mathbf{e}_{n-1} = [0, 0, 0, \ldots, 1, 0]^\top$. *The choice* $\mathbf{s}_1 = \mathbf{e}_n$, $\mathbf{s}_2 = \mathbf{e}_{n-1}$,..., $\mathbf{s}_{n-1} = \mathbf{e}_2$ *constitutes a* peel-up approach.

The above definition is motivated by the fact that the peel-down approach leads to the splitting in bordered matrices proposed in (Zanna & Munthe-Kaas 2002): the bordered matrices are obtained by 'peeling' the matrix Z from the top-left corner downwards. In the peel-up approach, we target rows and columns of Z starting from the bottom-right corner instead and proceed upwards.

In what follows, we apply a peel-up approach to (3.1) and discuss in detail the first stage, corresponding to the automorphism Ad_{S_1}, $\mathbf{s}_1 = \mathbf{e}_n$. The remaining stages of the peel-up approach share very similar features.

Using $\mathrm{Ad}_{\mathrm{diag}(-1)^{\mathbf{e}_n}}$ to perform the first algebra splitting we obtain $Z = P + K$, where

$$P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & 0 & \gamma_{n-1} \\ 0 & \cdots & 0 & \beta_{n-1} & 0 \end{bmatrix}, \; K = \begin{bmatrix} \alpha_1 & \gamma_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \gamma_{n-2} & 0 \\ \vdots & \ddots & \beta_{n-2} & \alpha_{n-1} & 0 \\ 0 & \cdots & 0 & 0 & \alpha_n \end{bmatrix}. \tag{3.3}$$

Next, we start computing commutators. We write

$$P = \gamma_{n-1}\mathbf{e}_{n-1}\mathbf{e}_n^\top + \beta_{n-1}\mathbf{e}_n\mathbf{e}_{n-1}^\top,$$

and commence our computations with

$$\begin{aligned}[P,K] = &-\gamma_{n-2}\gamma_{n-1}\mathbf{e}_{n-2}\mathbf{e}_n^\top + \beta_{n-2}\beta_{n-1}\mathbf{e}_n\mathbf{e}_{n-2}^\top \\ &+ \gamma_{n-1}(\alpha_n - \alpha_{n-1})\mathbf{e}_{n-1}\mathbf{e}_n^\top - \beta_{n-1}(\alpha_n - \alpha_{n-1})\mathbf{e}_n\mathbf{e}_{n-1}^\top.\end{aligned} \tag{3.4}$$

It is immediate to observe that there appears a fill-in in the tridiagonal structure of $X(t) = Pt - \frac{1}{2}t^2[P,K] + \mathcal{O}(t^3)$. In general, the more commutators we take, the greater the fill-in: two extra nonzero elements (one in the nth row and one in the nth column) for every extra power of t. In principle, the whole nth row and column would be eventually filled in. But this is not such bad news as it might appear at a first glance, and below we explain the reason.

3.1. Dealing with fill-in. An important observation is that fill-in of $X(t)$ propagates only in the Lie triple-system $\mathfrak{p}$, which consists of rank-2 matrices of the form

$$\left[\begin{array}{c|c} O & \mathbf{a} \\ \hline \mathbf{b}^\top & 0 \end{array}\right], \qquad \mathbf{a},\mathbf{b}\in\mathbb{R}^m. \tag{3.5}$$

Therefore, once $X(t)$ is approximated, its exponential can be computed exactly by means of an expression analogous to the Euler–Rodrigues formula for the exponential of a skew-symmetric matrix. Assume that $A\in\mathfrak{p}$ is of the form (3.5) Then,

$$\exp(A) = \begin{cases} I + \frac{\sinh\theta}{\theta}A + \frac{1}{2}\left[\frac{\sinh(\theta/2)}{\theta/2}\right]^2 A^2, & \mathbf{a}^\top\mathbf{b} > 0, \theta = \sqrt{\mathbf{a}^\top\mathbf{b}}, \\ I + A + \frac{1}{2}A^2, & \mathbf{a}^\top\mathbf{b} = 0, \\ I + \frac{\sin\theta}{\theta}A + \frac{1}{2}\left[\frac{\sin(\theta/2)}{\theta/2}\right]^2 A^2, & \mathbf{a}^\top\mathbf{b} < 0, \theta = \sqrt{-\mathbf{a}^\top\mathbf{b}} \end{cases} \tag{3.6}$$

(Zanna & Munthe-Kaas 2002), where

$$A^2 = \left[\begin{array}{c|c} \mathbf{a}\mathbf{b}^\top & \mathbf{0} \\ \hline \mathbf{0}^\top & \theta^2 \end{array}\right].$$

Setting to η_1,η_2 the coefficients of (3.6), application to a vector yields

$$\exp(A)\mathbf{v} = \mathbf{v} + \eta_1\begin{bmatrix} O & \mathbf{a} \\ \mathbf{b}^\top & 0 \end{bmatrix}\mathbf{v} + \eta_2\begin{bmatrix} \mathbf{a}\mathbf{b}^\top & \mathbf{0} \\ \mathbf{0}^\top & \theta \end{bmatrix}\mathbf{v}.$$

Assume next that $\mathbf{w},\mathbf{v}\in\mathbb{R}^{k+1}, \mathbf{a},\mathbf{b}\in\mathbb{R}^k$. Writing $\mathbf{v} = [\mathbf{v}_k, v]^\top$, $\mathbf{w} = [\mathbf{w}_k, w]^\top$, a direct computation reveals that

$$\begin{bmatrix} \mathbf{w}_k \\ w \end{bmatrix} = \exp(A)\mathbf{v} = \begin{bmatrix} \mathbf{v}_k + \zeta_1\mathbf{a} \\ \zeta_2 \end{bmatrix}, \tag{3.7}$$

where

$$\begin{aligned}\zeta_1 &= [\eta_1 v + \eta_2(\mathbf{b}^\top\mathbf{v}_k)], \\ \zeta_2 &= (1+\theta)v + (\mathbf{b}^\top\mathbf{v}_k).\end{aligned}$$

TABLE 3.1

Cost of the computation (including both addition and multiplication) of the exponential (3.6). The (k,k) column corresponds to the case when $\mathbf{a},\mathbf{b}$ are full, the (k,p) corresponds to the case when $\mathbf{a}$ is full while only the last p components of $\mathbf{b}$ are nonzero and finally the (p,p) column corresponds to both $\mathbf{a}$ and $\mathbf{b}$ having only the last p components nonzero.

Cost of $\exp(A)$	(k,k)	(k,p)	(p,p)
$\mathbf{a}^\top\mathbf{b}$	$2k$	$2p$	$2p$
$\mathbf{b}^\top\mathbf{v}_k$	$2k$	$2p$	$2p$
$\zeta_1\mathbf{a}$	k	k	p
$\mathbf{w}_k$	k	k	p
total, stage k	$6k$	$2k+4p$	$6p$
total, summing $1\le k\le n$ (vector)	$3n^2$	n^2+4pn	$6pn$
matrix (n vectors)	$2n^3$	n^3+2pn^2	$4pn^2$

When the exponential is applied to a matrix, we apply (3.7) to each column vector. Note, however, that the scalar product $\mathbf{a}^\top\mathbf{b}$ need be computed only once, even if the matrix columns are distinct.

In passing, we mention that there exist another formula for the exact computation of the exponential of a matrix as in (3.5), due to Celledoni & Iserles (2000),

$$\exp(A) = I + [\mathbf{e}_k, \mathbf{k}]\varphi(D)\begin{bmatrix} \mathbf{l}^\top \\ \mathbf{e}_k^\top \end{bmatrix}, \tag{3.8}$$

where

$$\mathbf{k} = \begin{bmatrix} \mathbf{a} \\ 0 \end{bmatrix}, \quad \boldsymbol{l} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}, \quad D = \begin{pmatrix} 0 & \mathbf{b}^\top\mathbf{a} \\ 1 & 0 \end{pmatrix},$$

$\mathbf{e}_k$ is the vector $[0,0,\ldots,0,1]^\top \in \mathbb{R}^k$ and finally $\varphi(z) = (\mathrm{e}^z-1)/z$. This formula can be shown to have the same computational cost as (3.7).

If $\mathbf{a},\mathbf{b}$ have $p \ll n$ nonzero elements only, say $a_{n-p},\ldots,a_{n-1},b_{n-p}\ldots,b_{n-1}$, it is clear that the computation of exponentials of elements in $\mathfrak{p}$ requires just $\mathcal{O}(pn)$ operations, when the exponentials are multiplied by a vector, and $\mathcal{O}\big(pn^2\big)$ operations when a multiplication by a matrix is required (see Table 3.1). Therefore, the fill-in in the $\mathfrak{p}$ part is inconvenient (in principle one would have preferred to preserve the neater tridiagonal structure), but it is not dangerous insofar as the increase of the computational cost is concerned.

More subtle is the case when the fill-in appears in $\mathfrak{k} \ni Y(t) = Kt - \frac{1}{12}t^3[P,[P,K]] +$ higher order terms. If such fill-in occurs and it is not suitably dealt with, it will propagate further and the tridiagonal structure of the submatrices will lost. The matrices then become increasingly fuller and fuller, and the cost of the splitting becomes $\mathcal{O}\big(n^3\big)$, as discussed in (Zanna & Munthe-Kaas 2002). Such an instance is displayed below in Figure 3.1: we perform three steps of the peel-up procedure for a tridiagonal symmetric matrix. We compute the function $Y(t)$ given in (2.5) and truncate the expansion to order six. It is clearly observed that in each step the number of non-zero elements increases and that the submatrices have a tendency to become full. This is clearly not desirable, since we do not want to lose the benefits of the original tridiagonal form!

However, two important observations are:

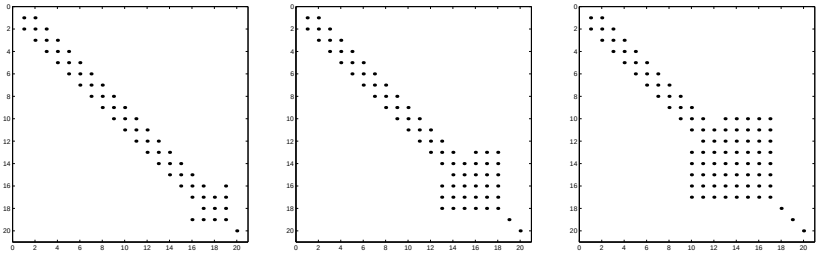


FIGURE 3.1. *Fill-in in the tridiagonal structure at step one, two and three in the peel-up procedure.*

- The fill-in in the $Y(t)$ part appears at order five only: Truncations of order one to four are still tridiagonal.
- The number of fill-in elements is usually very small.

Hence it is reasonable to eliminate the fill-in at each step by means of similarity orthogonal transformations (for instance, Givens rotations). At each step this contributes only $\mathcal{O}(1)$ to the cost of the splitting, which is negligible compared to the total cost of the approximation.

If Z is a skew-symmetric matrix, the fill-in in the $Y(t)$ part is chessboard-like. Givens rotations can be used to eliminate the subdiagonal fill-in and their transpose takes care of the superdiagonal fill-in. This is precisely the point when our choice of the peel-up approach, in preference to peel-down, starts to pay dividends. Using a peel-down approach, Givens rotations that eliminate the subdiagonal fill-in cause further fill-in that is propagated downwards. However, no fill-in is caused if Givens rotations are targeted to annihilate superdiagonal fill-ins instead (compare Figures (3.2) and (3.3)). For this reason we will restrict our attention to the peel-up approach instead of the peel-down approach of (Zanna & Munthe-Kaas 2002).

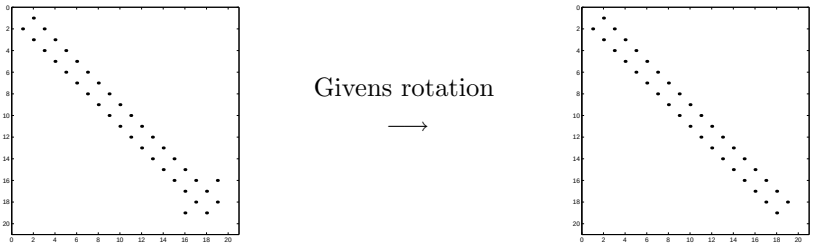


FIGURE 3.2. *In the peel-up approach for skew-symmetric matrices, using a Givens rotation to annihilate subdiagonal fill-in does not cause further fill-in.*

3.2. On the computation of commutators. We return to the computation of commutators and denote by $\mathrm{ad}_B = [B, \cdot\,]$ the operator that performs commutation with B, i.e. $\mathrm{ad}_B C = [B, C]$, $\mathrm{ad}_B^2 C = [B, [B, C]]$, et cetera.

Our first observation is that the involutions S are usually chosen so that $P = \Pi_{\mathfrak{p}}(Z)$ has low rank, hence only just a few nonzero eigenvalues. Thus, we can use the theory of minimal polynomials of matrices (Horn & Johnson 1985) so that few

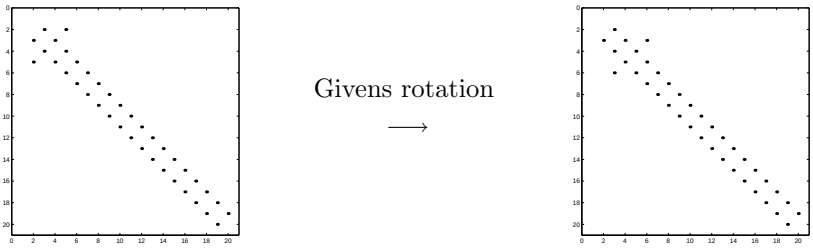


FIGURE 3.3. *In the peel-down approach for skew-symmetric matrices, using a Givens rotation to annihilate subdiagonal fill-in does causes fill-in that propagates downwards. If the rotations are chosen to annihilate* superdiagonal *fill-in instead, there is no fill-in propagation.*

commutators need be computed. All the remaining commutators with P can be obtained as linear combinations of those.

LEMMA 3.2. *Consider the matrix A of the form (3.5) with $\mathbf{a}\mathbf{b}^\top \neq O$. The minimal polynomial of* ad_A *is*

$$p(\lambda) = \lambda(\lambda - 2\theta)(\lambda + 2\theta)(\lambda - \theta)(\lambda + \theta) = \lambda^5 - 5\mathbf{b}^\top\mathbf{a}\lambda^3 + 4(\mathbf{b}^\top\mathbf{a})^2\lambda, \tag{3.9}$$

where $\theta = \sqrt{\mathbf{b}^\top\mathbf{a}}$.

If $\mathbf{a}\mathbf{b}^\top = O$, and neither $\mathbf{a}$ nor $\mathbf{b}$ is zero, then the minimal polynomial is

$$p(\lambda) = \lambda^3. \tag{3.10}$$

Proof. Recall that if A has distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ with algebraic multiplicities $r_1, r_2, \ldots, r_m$ respectively, the minimal polynomial of A has the form

$$q(\lambda) = \prod_{i=1}^{m} (\lambda - \lambda_i)^{g_i},$$

where g_i is the order of the largest Jordan block of A corresponding to the eigenvalue λ_i (Horn & Johnson 1985).

Let us assume first that $\mathbf{b}^\top\mathbf{a} \neq 0$. Imposing $A\mathbf{v} = \lambda\mathbf{v}$, we deduce immediately that the eigenvalues of A are $\lambda = \pm\theta = \pm\sqrt{\mathbf{b}^\top\mathbf{a}}$ and $\lambda = 0$ with algebraic multiplicities one, one, and $n-2$ respectively. It is easily verified that these are also their geometric multiplicities: for $\lambda = \pm\theta$, eigenvectors are of the form $[\mathbf{a}, \pm 1]^\top$; for the zero eigenvalues, eigenvectors are of the form $[\mathbf{v}_1, 0]^\top$, $\mathbf{0} \neq \mathbf{v}_1 \in \mathbb{R}^{n-1}$, satisfying $\mathbf{b}^\top\mathbf{v}_1 = 0$, furthermore, it is possible to find $n-2$ of those that are linearly independent.

Since the eigenvalues and eigenvectors of ad_A are the form $\lambda_i - \lambda_j$ and $\mathbf{y}_i^\top\mathbf{x}_j$ respectively, the λ_is being eigenvalues of A with left and right eigenvector $\mathbf{y}_i$ and $\mathbf{x}_i$ respectively, we deduce that ad_A has eigenvalues

$$\lambda = \pm 2\theta, \qquad \lambda = \pm\theta$$

with algebraic/geometric multiplicities one each, and

$$\lambda = 0$$

with algebraic and geometric multiplicity n^2-4. This implies that all Jordan blocks have size one, from which it follows directly that the minimal polynomial of ad_A is of the form (3.9).

Next, if $\theta = 0$ but $\mathbf{a}\mathbf{b}^\top \neq O$, namely $\mathbf{a}, \mathbf{b} \neq \mathbf{0}$, the eigenvalues of A, that we write as $[\mathbf{v}_1, v_2]^\top$, must obey the conditions

$$\begin{aligned} \mathbf{a}v_2 &= \mathbf{0} \\ \mathbf{b}^\top \mathbf{v}_1 &= 0. \end{aligned}$$

Since $\mathbf{a} \neq 0$, it must necessarily be $v_2 = 0$. Therefore eigenvalues must be of the form $[\mathbf{v}_1, 0]$. Recall that $\mathbf{v}_1$ has $n-1$ entries ($n-1$ free parameters) while the second equation $\mathbf{b}^\top \mathbf{v}_1 = 0$ gives only a linear constraint: This mean that we can find only $n-2$ linearly independent eigenvalues and two further linearly independent generalized eigenvalues. In terms of Jordan blocks, this means that A has a Jordan block of the form

$$J(0) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

hence λ^3 is the minimal polynomial of A and, as a consequence, $A^3 = O$. Passing to the adjoint operator ad_A, recall that, for an arbitrary matrix C,

$$\mathrm{ad}_A^k C = \sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} A^i C A^{k-i}, \qquad k = 1, 2, \ldots. \tag{3.11}$$

Clearly, $\mathrm{ad}_A^5 C = O$ since in all terms there appears a power A^i with $i \geq 3$. For lower order powers, there are always terms of the type $A^i C A^{k-i}$ where $i, k-i \leq 2$. This means that it is always possible to find a matrix C for which at least one of terms does not vanish. Hence the minimal polynomial of ad_A is

$$p(\lambda) = \lambda^5.$$

Finally, in the case when either $\mathbf{a} = \mathbf{0}$ or $\mathbf{b} = \mathbf{0}$, by direct computation,

$$A^2 = O,$$

hence the minimal polynomial of A is λ^2. Insofar as ad_A is concerned, the first power to vanish in (3.11) is ad_A^3, and no lower power vanishes for arbitrary matrices C. Hence the minimal polynomial is

$$p(\lambda) = \lambda^3.$$

This completes the proof of the lemma. □

THEOREM 3.3. *Assume that the matrix A is of the form (3.5). Then, for every $k = 1, 2, \ldots$, commutators by A can be computed as*

$$\mathrm{ad}_A^k = [C_1 + (-1)^k C_2] 2^k \theta^k + [C_3 + (-1)^k C_4] \theta^k, \qquad k = 1, 2, \ldots \tag{3.12}$$

when $\theta = \sqrt{\mathbf{b}^\top \mathbf{a}} \neq 0$, *and*

$$
\begin{aligned}
C_1 - C_2 &= \frac{1}{6}\left(-\frac{\mathrm{ad}_A}{\theta} + \frac{\mathrm{ad}_A^3}{\theta^3}\right) \\
C_3 - C_4 &= \frac{1}{3}\left(\frac{4\mathrm{ad}_A}{2\theta} - \frac{\mathrm{ad}_P^3}{\theta^3}\right) \\
C_1 + C_2 &= \frac{1}{12}\left(-\frac{\mathrm{ad}_A^2}{\theta^2} + \frac{\mathrm{ad}_A^4}{\theta^4}\right) \\
C_3 + C_4 &= \frac{1}{3}\left(\frac{4\mathrm{ad}_A^2}{\theta^2} - \frac{\mathrm{ad}_A^4}{\theta^4}\right).
\end{aligned} \tag{3.13}
$$

If $\theta = 0$ *but* $\mathbf{a}\mathbf{b}^\top \neq O$, *then*

$$\mathrm{ad}_A^k = O, \qquad k = 5, 6, 7, \ldots.$$

If $\theta = 0$ *and either* $\mathbf{a}$ *or* $\mathbf{b}$ *is a zero vector, then*

$$\mathrm{ad}_A^k = O, \qquad k = 3, 4, 5, \ldots.$$

Proof. Recall that the minimal polynomial is the least degree monic polynomial such that

$$p(\mathrm{ad}_A) = 0,$$

hence

$$\mathrm{ad}_A^5 - 5(\mathbf{b}^\top \mathbf{a})\mathrm{ad}_A^3 + 4(\mathbf{b}^\top \mathbf{a})^2 \mathrm{ad}_A = O.$$

Multiplying by ad_A, ad_A^2, …, we obtain the recurrence relation

$$\mathrm{ad}_A^{k+2} - 5(\mathbf{b}^\top \mathbf{a})\mathrm{ad}_A^k + 4(\mathbf{b}^\top \mathbf{a})^2 \mathrm{ad}_A^{k-2} = O, \qquad k = 3, 4, 5, \ldots,$$

whose general solution is (3.12). The unknowns C_1, C_2, C_3, C_4 are obtained by requiring that the formula (3.12) is correct for $k = 1, 2, 3, 4$. We obtain

$$
\begin{aligned}
(C_1 - C_2) + \frac{1}{2}(C_3 - C_4) &= \frac{1}{2\theta}\mathrm{ad}_A, \\
(C_1 + C_2) + \frac{1}{4}(C_3 + C_4) &= \frac{1}{4\theta^2}\mathrm{ad}_A^2, \\
(C_1 - C_2) + \frac{1}{8}(C_3 - C_4) &= \frac{1}{8\theta^3}\mathrm{ad}_A^3, \\
(C_1 + C_2) + \frac{1}{16}(C_3 + C_4) &= \frac{1}{16\theta^4}\mathrm{ad}_A^4,
\end{aligned}
$$

and (3.13) follows by direct computation. Thus, (3.12) is determined for both odd and even values of k. □

In other words, given an arbitrary matrix B, the commutator $\mathrm{ad}_A^k B$, $k \geq 5$, can be obtained as a mere linear combination of $B, \mathrm{ad}_A B, \ldots, \mathrm{ad}_A^4 B$. In our case, taking $A \equiv P$ and taking into account the sparsity of P, the computation of ad_P^k, $k = 1, 2, 3, 4$, is particularly simple. Setting $E_{i,j} = \mathbf{e}_i \mathbf{e}_j^\top$, a matrix with 1 in the (i, j)

position and 0 otherwise, and applying ad_P to K (whose elements are as in (3.3)), we have

$$\begin{aligned}
[P,K] &= c_{n-2,n}E_{n-2,n} + c_{n,n-2}E_{n,n-2} \\
&\quad + c_{n-1,n}E_{n-1,n} + c_{n,n-1}E_{n,n-1} \\
[P,[P,K]] &= d_{n-2,n-1}E_{n-2,n-1} + d_{n-1,n-2}E_{n-1,n-2} \\
&\quad + d_{n-1,n-1}E_{n-1,n-1} + d_{n,n}E_{n,n} \\
[P,[P,[P,K]]] &= e_{n-2,n}E_{n-2,n} + e_{n,n-2}E_{n,n-2} \\
&\quad + e_{n-1,n}E_{n-1,n} + e_{n,n-1}E_{n,n-1} \\
[P,[P,[P,[P,K]]]] &= f_{n-2,n-1}E_{n-2,n-1} + f_{n-1,n-2}E_{n-1,n-2} \\
&\quad + f_{n-1,n-1}E_{n-1,n-1} + f_{n,n}E_{n,n}.
\end{aligned} \tag{3.14}$$

The nonzero coefficients $c_{i,j}, d_{i,j}, e_{i,j}, f_{i,j}$ are given by

$$\begin{aligned}
c_{n-2,n} &= -\gamma_{n-2}\gamma_{n-1} \\
c_{n,n-2} &= \beta_{n-2}\beta_{n-1} \\
c_{n-1,n} &= \gamma_{n-1}(\alpha_n - \alpha_{n-1}) \\
c_{n,n-1} &= \beta_{n-1}(\alpha_n - \alpha_{n-1}) \\
\\
d_{n-2,n-1} &= -\beta_{n-1}c_{n-2,n} \\
d_{n-1,n-2} &= \gamma_{n-1}c_{n,n-2} \\
d_{n-1,n-1} &= \gamma_{n-1}c_{n,n-1} - \beta_{n-1}c_{n-1,n} \\
d_{n,n} &= -d_{n-1,n-1} \\
\\
e_{n-2,n} &= -\gamma_{n-1}d_{n-2,n-1} \\
e_{n,n-2} &= \beta_{n-1}d_{n-1,n-2} \\
e_{n-1,n} &= 2\gamma_{n-1}d_{n,n} \\
e_{n,n-1} &= -2\beta_{n-1}d_{n-1,n-1} \\
\\
f_{n-2,n-1} &= -\beta_{n-1}e_{n-2,n} \\
f_{n-1,n-2} &= \gamma_{n-1}e_{n,n-2} \\
f_{n-1,n-1} &= \gamma_{n-1}e_{n,n-1} - \beta_{n-1}e_{n-1,n} \\
f_{n,n} &= -f_{n-1,n-1},
\end{aligned} \tag{3.15}$$

where β_ks and γ_ks originate in (3.3).

Unfortunately, the theory of minimal polynomials is not equally insightful insofar as commutators with K are concerned. Instead, we have computed the first few such

terms explicitely and they are also in a form that renders their evaluation cheap,

$$
\begin{aligned}
[K,[P,K]] &= g_{n,n-3}E_{n,n-3} + g_{n,n-2}E_{n,n-2} + g_{n,n-1}E_{n,n-1} \\
&\quad + g_{n-3,n}E_{n-3,n} + g_{n-2,n}E_{n-2,n} + g_{n-1,n}E_{n-1,n} \\
[K,[K,[P,K]]] &= h_{n,n-4}E_{n,n-4} + h_{n,n-3}E_{n,n-3} + h_{n,n-2}E_{n,n-2} \\
&\quad + h_{n,n-1}E_{n,n-1} + h_{n-4,n}E_{n-4,n} + h_{n-3,n}E_{n-3,n} \\
&\quad + h_{n-2,n}E_{n-2,n} + h_{n-1,n}E_{n-1,n} \\
[K,[P,[P,[P,K]]]] &= i_{n,n-3}E_{n,n-3} + i_{n,n-2}E_{n,n-2} + i_{n,n-1}E_{n,n-1} \\
&\quad + i_{n-3,n}E_{n-3,n} + i_{n-2,n}E_{n-2,n} + i_{n-1,n}E_{n-1,n} \\
[K,[K,[K,[P,K]]]] &= j_{n,n-5}E_{n,n-5} + j_{n,n-5}E_{n,n-4} + j_{n,n-3}E_{n,n-3} \\
&\quad + j_{n,n-2}E_{n,n-2} + j_{n,n-1}E_{n,n-1} \\
&\quad + j_{n-5,n}E_{n-5,n} + j_{n-4,n}E_{n-4,n} + j_{n-3,n}E_{n-3,n} \\
&\quad + j_{n-2,n}E_{n-2,n} + j_{n-1,n}E_{n-1,n} \\
[[P,K],[P,[P,K]]] &= k_{n,n-2}E_{n,n-2} + k_{n,n-1}E_{n,n-1} \\
&\quad + k_{n-2,n}E_{n-2,n} + k_{n-1,n}E_{n-1,n} \\
[K,[K,[P,[P,K]]]] &= l_{n-1,n-4}E_{n-1,n-4} + l_{n-1,n-3}E_{n-1,n-3} \\
&\quad + l_{n-1,n-2}E_{n-1,n-2} + l_{n-1,n-1}E_{n-1,n-1} \\
&\quad + l_{n-2,n-3}E_{n-2,n-3} + l_{n-2,n-2}E_{n-2,n-2} \\
&\quad + l_{n-2,n-1}E_{n-2,n-1} + l_{n-3,n-2}E_{n-3,n-2} \\
&\quad + l_{n-3,n-1}E_{n-3,n-1} + l_{n-4,n-1}E_{n-4,n-1} \\
[[P,K],[K,[P,K]]] &= m_{n,n}E_{n,n} + m_{n-1,n-3}E_{n-1,n-3} \\
&\quad + m_{n-1,n-2}E_{n-1,n-2} + m_{n-1,n-1}E_{n-1,n-1} \\
&\quad + m_{n-2,n-3}E_{n-2,n-3} + m_{n-2,n-2}E_{n-2,n-2} \\
&\quad + m_{n-2,n-1}E_{n-2,n-1} + m_{n-3,n-2}E_{n-3,n-2} \\
&\quad + m_{n-3,n-1}E_{n-3,n-1}.
\end{aligned}
\tag{3.16}
$$

The nonzero coefficients of (3.16) are reported in the appendix.

Note that, when Z in (3.1) is symmetric or skew-symmetric, only about a half of the coefficients in (3.14) and (3.16) need be computed. To see this, assume that Z is symmetric and so are P and K. Then, so is $[P,[P,K]],[K,[P,K]],\ldots$ and in general all terms which include an even number of commutators. Those including an odd number of commutators, like $[P,K],[K,[K,[P,K]]],\ldots$, are instead skew-symmetric. Symmetry and skew-symmetry can be transparently taken into account when computing the coefficients (3.15) and (A.1)–(A.6).

By a similar token, when Z is skew-symmetric, so are P and K, and *all* their commutators. Again, this means that we only need to compute just the under-diagonal (or over-diagonal) coefficients in (3.14) and (3.16).

3.3. The reduction to a tridiagonal form: Symmetric and skew-symmetric matrices. For symmetric and skew-symmetric matrices, the problem of reduction to a tridiagonal form is classical, and we briefly review well-known techniques based on Householder reflections and Lanczos tridiagonalisation.

Symmetric and skew-symmetric matrices can be reduced to a tridiagonal form by means of Householder reflections,

$$H = I - \beta \mathbf{v}\mathbf{v}^\top, \qquad \beta = \frac{2}{\|\mathbf{v}\|^2},$$

which are orthogonal transformations. Assume that Z is a full matrix, and let

$$\mathbf{v} = \begin{bmatrix} 0 \\ Z_{2,1} \pm \|\mathbf{z}_1\| \\ Z_{3,1} \\ \vdots \\ Z_{n,1} \end{bmatrix},$$

where $\mathbf{z}_1$ denotes the first column of Z. Then $H = H^\top$ and it can be easily verified that

$$\begin{array}{rcl} Z \text{ symmetric} & \Rightarrow & HZH \text{ symmetric} \\ Z \text{ skew-symmetric} & \Rightarrow & HZH \text{ skew-symmetric.} \end{array}$$

For simplicity, let us rewrite the matrix H as $I - \mathbf{w}\mathbf{w}^\top$, where $\|\mathbf{w}\|^2 = 2$. We have

$$HZH = Z - (\mathbf{w}\mathbf{w}^\top Z + Z\mathbf{w}\mathbf{w}^\top) + (\mathbf{w}^\top Z\mathbf{w})\mathbf{w}\mathbf{w}^\top. \tag{3.17}$$

The above transformation can be computed very effectively for symmetric and skew-symmetric matrices thanks to an algorithm due to Wilkinson: the main idea is to split the quadratic term in (3.17) into two terms that are subsumed in the computation of the second and third term instead.

In more detail,

1. Compute $\mathbf{g} = Z\mathbf{w}$ and $\alpha = \frac{1}{2}\mathbf{w}^\top\mathbf{g}$. Note that $\alpha = 0$ when Z is skew-symmetric!
2. Replace $\mathbf{g}$ by $\mathbf{g} - \alpha\mathbf{h}$ (this step is skipped in the skew-symmetric case).
3. Compute $HZH = Z - \mathbf{g}\mathbf{w}^\top - (\mathbf{g}\mathbf{w}^\top)^\top$ when Z is symmetric, or $HZH = Z - \mathbf{g}\mathbf{w}^\top + (\mathbf{g}\mathbf{w}^\top)^\top$ when Z is skew-symmetric.

Ignoring lower-order terms, the computation of HZH reduces to $3(n-1)^2$ operations, counting both additions and multiplications. Of those, $2(n-1)^2$ arise from the matrix-vector product at step 1, while the remaining $(n-1)^2$ from the matrix update at step 3. The remaining steps have lower cost, that we overlook. To reduce Z to a tridiagonal form, we have $n-1$ transformations on matrices of decreasing dimension $n \times n, (n-1) \times (n-1), \ldots, 3 \times 3$, resulting in

$$\sum_{k=3}^{n} 3(k-1)^2 \approx n^3$$

operations. If counting only multiplications (or only *flops*, i.e. operations of the form $av + b$), the count reduces to $\frac{2}{3}n^3$, consistently with (Golub & van Loan 1989).

When Z is sparse and very large, a possible alternative is to use the Lanczos method, which is particularly attractive when it is cheap to form products of the

form $Z\mathbf{v}$, where $\mathbf{v} \in \mathbb{R}^n$. Set

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & \cdots & 0 & \beta_{n-1} & \alpha_n \end{bmatrix},$$

when Z is symmetric, and

$$T = \begin{bmatrix} 0 & \beta_1 & 0 & \cdots & 0 \\ -\beta_1 & 0 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\beta_{n-2} & 0 & \beta_{n-1} \\ 0 & \cdots & 0 & -\beta_{n-1} & \end{bmatrix}$$

when Z is skew-symmetric, and denote by $Q = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n]$ an orthogonal matrix that tridiagonalizes Z, i.e. $Q^\top ZQ = T$, where T is as above. From

$$ZQ = QT,$$

we readily obtain the recurrences

$$Z\mathbf{q}_j = \beta_{j-1}\mathbf{q}_{j-1} + \alpha_j\mathbf{q}_j + \beta_j\mathbf{q}_{j+1} \tag{3.18}$$

$$Z\mathbf{q}_j = \beta_{j-1}\mathbf{q}_{j-1} - \beta_j\mathbf{q}_{j+1} \tag{3.19}$$

$j = 1, 2, \ldots, n-1$ for symmetric and skew-symmetric Z respectively, where $b_0 = 0$ and $\mathbf{q}_1 \in \mathbb{R}^n$ is an arbitrary vector such that $\|\mathbf{q}_1\| = 1$. Denoting, as usual, the residual as

$$\mathbf{r}_j = Z\mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1} - \alpha_j\mathbf{q}_j,$$
$$\mathbf{r}_j = Z\mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1},$$

for symmetric and skew-symmetric Z respectively, the coefficients α_j and β_j and the orthogonal vectors $\mathbf{q}_j$ in(3.18)-(3.19) can be computed as

$$\alpha_j = \mathbf{q}_j^\top Z\mathbf{q}_j, \quad \mathbf{q}_{j+1} = \frac{\mathbf{r}_{j+1}}{\|\mathbf{r}_j\|}, \quad \beta_j = \mathbf{q}_{j+1}^\top Z\mathbf{q}_j, \tag{3.20}$$

$$\beta_j = \|\mathbf{r}_j\|, \quad \mathbf{q}_{j+1} = \frac{\mathbf{r}_j}{\beta_j}, \tag{3.21}$$

respectively.

Both procedures break down when $\beta_j = 0$, which (in exact arithmetics) implies that T is reducible. In floating-point arithmetics it is also possible that the vectors $\mathbf{q}_j$ might become progressively less orthogonal. In such cases, a restart process is recommended (Golub & van Loan 1989, Saad 1992).

TABLE 3.2
Comparison of cost of the approximation of the exponential without (ZMK) and with reduction to tridiagonal form (IZ) for splittings of order $2, 3, 4$. *Only dominant terms are reported.*

Order 2	ZMK		IZ	
	vector	matrix	vector	matrix
Tridiag.	–	–	n^3	n^3
order cond.	$\frac{2}{3}n^3$	$\frac{2}{3}n^3$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
assembly exp	$3n^2$	$2n^3$	$6pn$	$4pn^2$
total	$\frac{2}{3}n^3$	$2\frac{2}{3}n^3$	$n^3+\mathcal{O}(pn)$	n^3+4pn^2

Order 3	ZMK		IZ	
	vector	matrix	vector	matrix
Tridiag.	–	–	n^3	n^3
order cond.	$2\frac{1}{2}n^3$	$2\frac{1}{2}n^3$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
assembly exp	$3n^2$	$2n^3$	$6pn$	$4pn^2$
total	$2\frac{1}{2}n^3$	$4\frac{1}{2}n^3$	$n^3+\mathcal{O}(pn)$	n^3+4pn^2

Order 4	ZMK		IZ	
	vector	matrix	vector	matrix
Tridiag.	–	–	n^3	n^3
order cond.	$4n^3$	$4n^3$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
assembly exp	$3n^2$	$2n^3$	$6pn$	$4pn^2$
total	$4n^3$	$6n^3$	$n^3+\mathcal{O}(pn)$	n^3+4pn^2

4. Other matrices. When Z is neither symmetric nor skew-symmetric, and, in particular, when it is not normal, the tridiagonalisation process (non-symmetric tridiagonalisation, i.e. tridiagonalisation by similarity transform, not necessarily orthonormal) might be either unstable or it might destroy the underlying algebraic structure (Golub & van Loan 1989). For instance, the tridiagonalisation of a matrix in the symplectic algebra $\mathfrak{sp}(n) := \{Z : ZJ = -JZ^\top\}$, where

$$J = \begin{bmatrix} O & I \\ -I & O \end{bmatrix}, \tag{4.1}$$

might not produce an output in $\mathfrak{sp}(n)$, and this is not desirable in many applications, for instance when conservation of Lie-group structure is important. To force the group structure, it might be more appropriate to look for other sparsity patterns that are (a) compatible with the algebra structure and (b) retained under commutation.

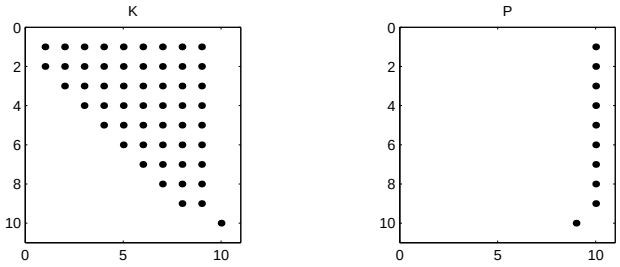
For matrices in $\mathfrak{gl}(n), \mathfrak{sl}(n)$ which do not strictly belong to other subalgebras, it is more convenient to reduce to an *upper Hessenberg form*, by means of orthogonal transformations (e.g., Householder reflections). This is a more stable process than non-symmetric tridiagonalisation (Golub & van Loan 1989). For generic matrices, reduction to upper Hessenberg form costs about $3\frac{1}{3}n^3$ operations (Golub & van Loan 1989).

For matrices belonging to other subalgebras, like the symplectic algebra, Lorentz-type algebras, quadratic algebras et cetera, it is possible to consider other specific

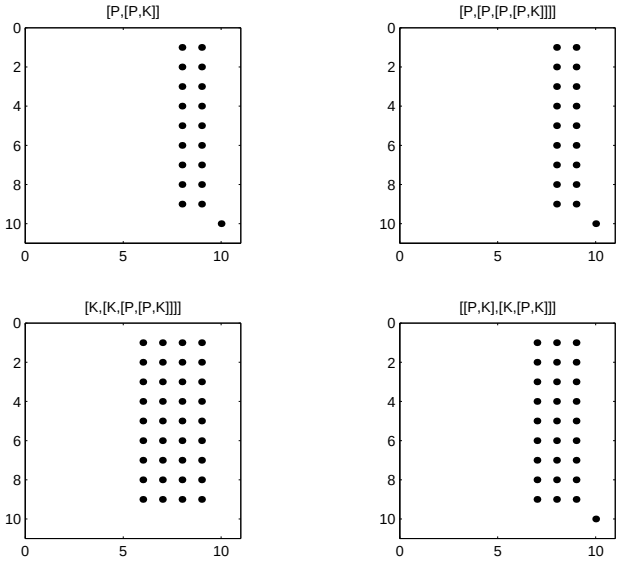
transformations that preserve the algebraic structure. These will be described at length in the sequel.

4.1. Upper Hessenberg matrices. In this subsection we analyse the first step of a peel-up approach, as in Definition 3.1, corresponding to the automorphism Ad_{S_1}, where $S_1 = \mathrm{diag}(-1)^{\mathbf{s}_1}, \mathbf{s}_1 = \mathbf{e}_n$. The remaining steps, corresponding to $\mathrm{Ad}_{S_2}, \mathrm{Ad}_{S_3}, \ldots$, corresponding to $\mathbf{s}_2 = \mathbf{e}_{n-1}$, $\mathbf{s}_3 = \mathbf{e}_{n-2}$, $\ldots$, share similar features.

Assume that Z in $\mathfrak{gl}(n)$ or in $\mathfrak{sl}(n)$ is in an upper Hessenberg form. The matrices P, K corresponding to the splitting induced by $\mathrm{Ad}_{\mathrm{diag}(-1)^{\mathbf{e}_n}}$ have the sparsity pattern



Computing the terms required in the generation of Y up to $\mathcal{O}(t^6)$ we have



therefore Y_6 has the sparsity pattern displayed in Figure 4.1 with just three filled-in entries, at the $(n-2, n-4)$, $(n-1, n-4)$ and $(n-1, n-3)$ positions. It thus takes just three Givens rotations to bring Y_6 to the same sparsity pattern as K via similarity transformations.

Next, we proceed to generate X. All the terms up to $\mathcal{O}(t^5)$ are

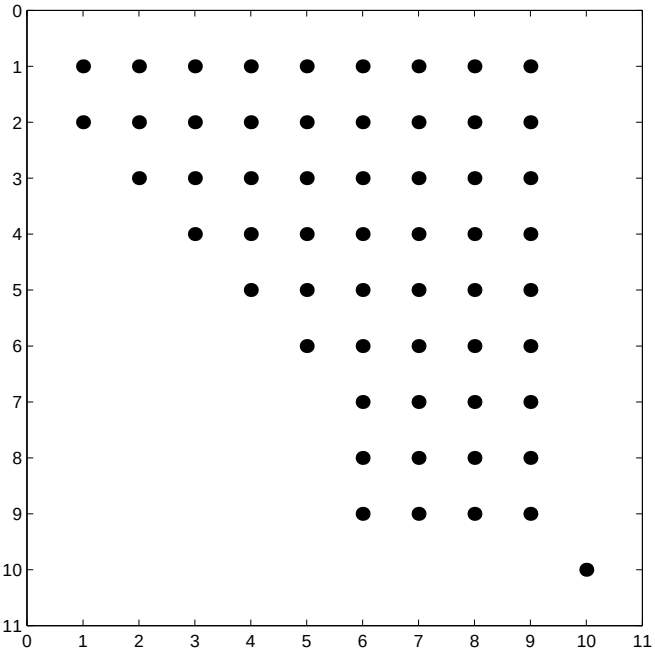
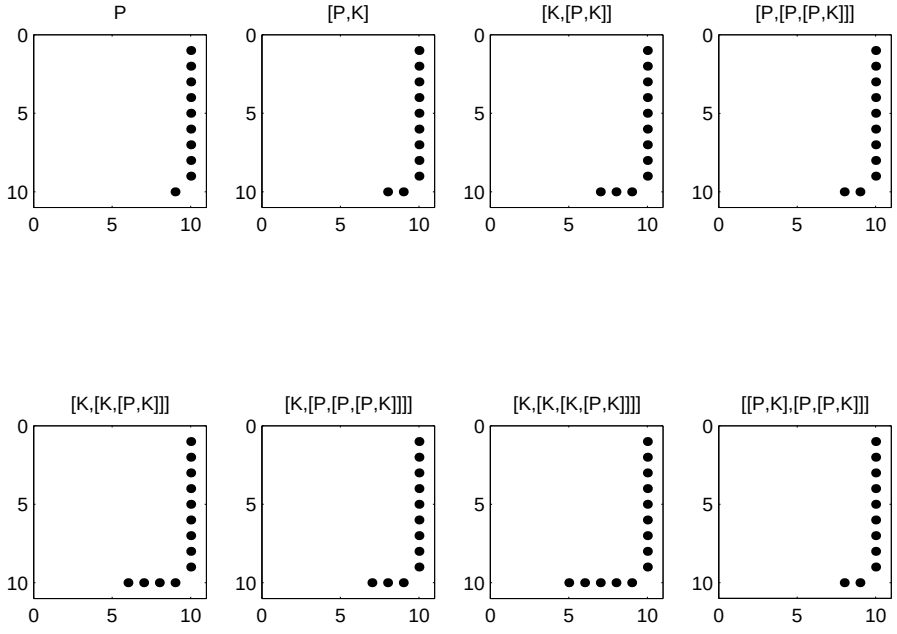
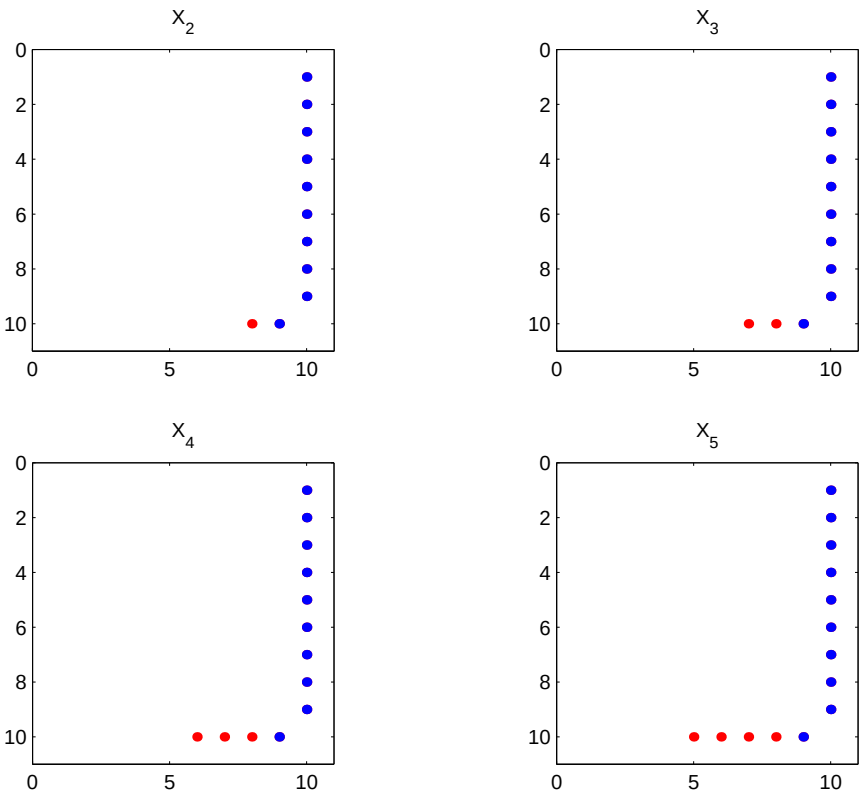


FIGURE 4.1. *Sparsity pattern of Y_6 for a matrix in a Hessenberg form.*



Therefore, the sparsity patterns of the truncations X_2, X_3, X_4, X_5 are



What about the exponential of X_p? Each such matrix is again of the form (3.5) and (3.6) still holds. Assume now that $\mathbf{a}$ has n nonzero elements, while $\mathbf{b}$ has only p nonzero elements, say $b_{n-p}, \ldots, b_{n-1}$. As displayed in Table 3.1, multiplying a vector by $n-1$ exponentials of matrices of decreasingly small dimension costs just $\mathcal{O}(pn^2)$ flops, while multiplying a matrix in a similar fashion carries the price tag of $\frac{1}{2}n^3 + \mathcal{O}(n^2)$ flops.

What is the cost of computing the commutators? Clearly, one has to take advantage of the sparsity of the matrices under consideration.

We commence with the analysis of commutators of the type $[A, B]$, where $A \in \mathfrak{p}$ and $B \in \mathfrak{k}$. We write

$$A = \begin{bmatrix} O & \mathbf{a} \\ \mathbf{b}^\top & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} B_1 & \mathbf{0} \\ \mathbf{0}^\top & B_2 \end{bmatrix},$$

where $\mathbf{b}^\top = [0, 0, \ldots, b_{n-q}, \ldots, b_{n-1}]$, has only q nonzero elements, B_1 is $(n-1) \times (n-1)$ and in an Hessenberg form, while B_2 has a single nonzero entry. We have

$$[A, B] = \begin{bmatrix} O & \mathbf{a}B_2 - B_1\mathbf{a} \\ \mathbf{b}^\top B_1 - B_2\mathbf{b}^\top & 0 \end{bmatrix}.$$

Thus, computing $[A, B]$ amounts to

- about $n(n-1)$ operations for the computation of $B_1\mathbf{a}$, since B_1 is in a Hessenberg form. Note that this reduces to about $2qn$ operations if only the last q columns of B_1 are nonzero.
- $(n-1)$ operations for $\mathbf{a}B_2$ and further $n-1$ operations to compute $\mathbf{a}B_2 - B_1\mathbf{a}$,

- about $q(q-1)$ operations for $\mathbf{b}^\top B_1$, q to compute $B_2\mathbf{b}$ and further q to compute $\mathbf{b}^\top B_1 - B_2\mathbf{b}^\top$.

In total, we require about n^2 operations (assuming that $q \ll n$). Running the commutators over matrices of decreasing dimension, we have in total

$$\sum_{j=1}^{n-1} j^2 \approx \frac{1}{3}n^3$$

operations. This is the cost of the commutators $[P,K], [K,[P,K]], [K,[K,[P,K]]], \ldots$, namely of $\mathfrak{p}$-elements with K.

Next, we consider commutators of the form $[A_1, A_2]$, where $A_1, A_2 \in \mathfrak{p}$ are bordered matrices. Set

$$A_1 = \begin{bmatrix} O & \mathbf{a} \\ \mathbf{b}^\top & 0 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} O & \mathbf{c} \\ \mathbf{d}^\top & 0 \end{bmatrix},$$

where only the last q elements of $\mathbf{b}, \mathbf{d}$ are nonzero. Since

$$[A_1, A_2] = \begin{bmatrix} \mathbf{a}\mathbf{c}^\top - \mathbf{d}\mathbf{b}^\top & \mathbf{0} \\ \mathbf{0}^\top & \mathbf{b}^\top\mathbf{d} - \mathbf{c}^\top\mathbf{a} \end{bmatrix},$$

it is evident that the computation of these commutators costs about $3q(n-1)$ operations, contributing a total of $\frac{3}{2}qn^2$ to the total cost (i.e. summing the contribution of similar terms over matrices of decreasing dimension). This is the cost of the commutators $[P,[P,K]], [P,[P,[P,[P,K]]]]$, etc. Finally, if $C_1, C_2 \in \mathfrak{k}$, with

$$C_1 = \begin{bmatrix} B_1 & \mathbf{0} \\ \mathbf{0}^\top & b_1 \end{bmatrix}, \qquad C_2 = \begin{bmatrix} B_2 & \mathbf{0} \\ \mathbf{0}^\top & b_2 \end{bmatrix},$$

with B_1, B_2 of dimension n, one has

$$[C_1, C_2] = \begin{bmatrix} [B_1, B_2] & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}.$$

If only the last p (respectively, r) columns of B_1, (resp. B_2) are nonzero, setting $r = \min\{p,q\}+1$, we deduce that the commutators $[C_1, C_2]$ cost about $6rn$ operations—an $\mathcal{O}\big(n^2\big)$ contribution when the count is carried over matrices of decreasing dimension.

Putting all the bricks together,

- Terms for order 2:

$$[P,K] \to \frac{1}{3}n^3$$

- Order 3:

$$[K,[P,K]] \to \frac{1}{3}n^3$$
$$[P,[P,K]] \to \mathcal{O}\big(n^2\big)$$

- Order 4:

$$[P,[P,[P,K]]] \to \mathcal{O}\big(n^2\big)$$
$$[K,[K,[P,K]]] \to \frac{1}{3}n^3$$

TABLE 4.1
Comparison of cost of the approximation of the exponential without (ZMK) and with reduction to Hessenberg form (IZ) for splittings of order $2, 3, 4$. *Only dominant terms are reported.*

Order 2	ZMK		IZ	
	vector	matrix	vector	matrix
Hessenberg	–	–	$3\frac{1}{3}n^3$	$3\frac{1}{3}n^3$
order cond.	$1\frac{1}{3}n^3$	$1\frac{1}{3}n^3$	$\frac{1}{3}n^3$	$\frac{1}{3}n^3$
assembly exp	$3n^2$	$2n^3$	n^2	n^3
total	$1\frac{1}{3}n^3$	$2\frac{1}{3}n^3$	$3\frac{2}{3}n^3$	$4\frac{2}{3}n^3$

Order 3	ZMK		IZ	
	vector	matrix	vector	matrix
Hessenberg	–	–	$3\frac{1}{3}n^3$	$3\frac{1}{3}n^3$
order cond.	$5n^3$	$5n^3$	$\frac{2}{3}n^3$	$\frac{2}{3}n^3$
assembly exp	$3n^2$	$2n^3$	n^2	n^3
total	$5n^3$	$7n^3$	$4n^3$	$5n^3$

Order 4	ZMK		IZ	
	vector	matrix	vector	matrix
Hessenberg	–	–	$3\frac{1}{3}n^3$	$3\frac{1}{3}n^3$
order cond.	$7n^3$	$7n^3$	n^3	n^3
assembly exp	$3n^2$	$2n^3$	n^2	n^3
total	$7n^3$	$9n^3$	$4\frac{1}{3}n^3$	$5\frac{1}{3}n^3$

- Order 5:

$$
\begin{aligned}
[K,[P,[P,[P,K]]]] &\to \frac{1}{3}n^3\\
[K,[K,[K,[P,K]]]] &\to \frac{2}{3}n^3\\
[[P,K],[P,[P,K]]] &\to \mathcal{O}\big(n^2\big)\\
[P,[P,[P,[P,K]]]] &\to \mathcal{O}\big(n^2\big)\\
[[P,K],[K,[P,K]]] &\to \mathcal{O}\big(n^2\big)\\
[K,[K,[P,[P,K]]]] &\to \mathcal{O}\big(n^2\big)\,.
\end{aligned}
$$

In Table 4.1 we summarise the cost for the various stages of the exponential approximation of a generic matrix $Z \in \mathfrak{gl}(n)$ for orders $2, 3$ and 4 and compare our new algorithms with those proposed in (Zanna & Munthe-Kaas 2002). For full matrices, it is evident that the benefits of our approach appear for orders greater than two. For order four our the new algorithm is almost 40% faster than the one without transformation to an upper Hessenberg form.

4.2. Symplectic matrices. The symplectic group of matrices $\mathrm{Sp}(n)$ is the set of all invertible matrices M of dimension $2n$ such that $MJM^\top = J$ where J is as in (4.1). The symplectic algebra $\mathfrak{sp}(n)$ is the set of $2n \times 2n$ matrices such that $NJ = -JN^\top$.

A symplectic matrix $M \in \mathrm{Sp}(n)$ is said to be in a *butterfly form* if it can be written as

$$M = \begin{bmatrix} D_1 & B_1 \\ D_2 & B_2 \end{bmatrix},$$

where D_1, D_2 are $n \times n$ diagonal matrices and B_1, B_2 are $n \times n$ tridiagonal matrices. This butterfly form was considered by Benner, Faßbender & Watkins (1999) as a starting point of a QR-type algorithm (the SR-algorithm) to compute the eigenvalues of a symplectic matrix so that the transformed matrix remains symplectic at each step.

The transformation of a given symplectic matrix to a butterfly form can be performed by the use of three different types of similarity mappings with the following matrices,

- **symplectic Givens transformations:**,

$$G = \left[\begin{array}{ccc|ccc} I_{k-1} & & & & & \\ & c & & & s & \\ & & I_{n-k} & & & \\ \hline & & & I_{k-1} & & \\ & -s & & & c & \\ & & & & & I_{n-k} \end{array}\right],$$

- **symplectic Householder transformations:**

$$H = \left[\begin{array}{cc|cc} I_{k-1} & & & \\ & Q & & \\ \hline & & I_{k-1} & \\ & & & Q \end{array}\right], \qquad Q = I_{n-k+1} - \beta \mathbf{v}\mathbf{v}^\top, \quad \beta = \frac{2}{\|v\|^2},$$

- **symplectic Gauss transformations:**

$$L = \left[\begin{array}{cccc|cccc} I_{k-2} & & & & & & & \\ & c & & & & & d & \\ & & c & & & d & & \\ & & & I_{n-k} & & & & \\ \hline & & & & I_{k-2} & & & \\ & & & & & c^{-1} & & \\ & & & & & & c^{-1} & \\ & & & & & & & I_{n-k} \end{array}\right].$$

The following algorithm for reduction to a butterfly form is described in more detail in (Faßbender 2000):

Given a $2n \times 2n$ symplectic matrix M, compute its reduction to a butterfly form. M will be overwritten by its butterfly form.

```
for j = 1 : n − 1
  for k = n : −1 : j + 1
    compute G_k such that (G_k M)_{k+n,j} = 0
    M = G_k M G_k^⊤
  end
  if j < n − 1 then
    compute H_j such that (H_j M)_{j+2:n,j} = 0
```

```
    M = H_j M H_j^⊤
  end
  compute L_{j+1} such that (L_{j+1} M)_{j+1,j} = 0
  M = L_{j+1} M L_{j+1}^{-1}
  for k = n : -1 : j+1
    compute G_k such that (M G_k)_{j,k} = 0
    M = G_k^⊤ M G_k
  end
  if j < n-1 then
    compute H_j such that (M H_j)_{j,j+2+n:2n} = 0
    M = H_j^⊤ M H_j
  end
end
```

The algorithm introduces zeros in the rows by applying one of the above-mentioned transformations from the right, while zeros in the columns are obtained by applying the transformations from the left. To maintain similarity, the inverse of each transformation is applied also on the other side. The basic idea of the algorithm is, at each step j, (i) to bring the jth column of M into the desired form; (ii) to bring the $(n+j)$th row of M into the desired form. For more details, see (Faßbender 2000).

Clearly, symplectic transformations can also be used at the algebra level: our idea is to reduce a symplectic matrix $A \in \mathfrak{sp}(n)$ to a butterfly form, using the same algorithm as above. Note that

$$A \in \mathfrak{sp}(n) \Leftrightarrow A = \begin{bmatrix} A_1 & A_2 \\ A_3 & -A_1^\top, \end{bmatrix}$$

where A_1, A_2, A_3 are $n \times n$ matrices and A_2, A_3 are symmetric, therefore, a butterfly matrix $B \in \mathfrak{sp}(n)$ must be of the form

$$B = \begin{bmatrix} B_1 & B_2 \\ B_3 & -B_1^\top, \end{bmatrix}$$

where B_1 and B_3 are now diagonal, and B_2 is tridiagonal and symmetric.

Assume next that $B \in \mathfrak{sp}(n)$ is in a butterfly form. To approximate $\exp(tB)$ we use again a peel-up approach, however, the matrices S_i need be appropriately modified to preserve the $\mathfrak{sp}(n)$ structure. Set

$$\tilde{S}_i = \mathrm{diag}(-1)^{\tilde{\mathbf{s}}_i}, \quad \tilde{\mathbf{s}}_i = [s_{i;1}, \ldots, s_{i;n}, s_{i;1}, \ldots, s_{i;n}]. \tag{4.2}$$

In other words, the $\tilde{S}_i$s can be taken as direct products $S_i \times S_i$ of the matrices S_i in (2.11), and they act in the same manner on the first and the second n rows and columns of the matrix B.

To have a mental picture of the splitting induced by the automorphisms $\mathrm{Ad}_{\tilde{S}_i}$, we set $i = n$ and obtain subspaces $\mathfrak{p}$ and $\mathfrak{k}$ with the sparsity structure

$$\mathfrak{p} \ni \begin{bmatrix} 0 & \cdots & 0 & \times & 0 & \cdots & 0 & \times \\ \vdots & & \vdots & \times & \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & \times & 0 & \cdots & 0 & \times \\ \times & \times & \times & 0 & \times & \times & \times & 0 \\ 0 & \cdots & 0 & \times & 0 & \cdots & 0 & \times \\ \vdots & & \vdots & \times & \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & \times & 0 & \cdots & 0 & \times \\ \times & \times & \times & 0 & \times & \times & \times & 0 \end{bmatrix}, \quad \mathfrak{k} \ni \begin{bmatrix} \times & \cdots & \times & 0 & \times & \cdots & \times & 0 \\ \vdots & & \vdots & 0 & \vdots & & \vdots & \vdots \\ \times & \cdots & \times & 0 & \times & \cdots & \times & 0 \\ 0 & 0 & 0 & \times & 0 & 0 & 0 & \times \\ \times & \cdots & \times & 0 & \times & \cdots & \times & 0 \\ \vdots & & \vdots & 0 & \vdots & & \vdots & \vdots \\ \times & \cdots & \times & 0 & \times & \cdots & \times & 0 \\ 0 & 0 & 0 & \times & 0 & 0 & 0 & \times \end{bmatrix}$$

respectively. Observe that the automorphism targets the $2n$th and nth rows and columns.

The sparsity pattern of the computed terms in the generation of $X(t)$ up to $\mathcal{O}\big(t^5\big)$ is displayed in Figure 4.2 below.

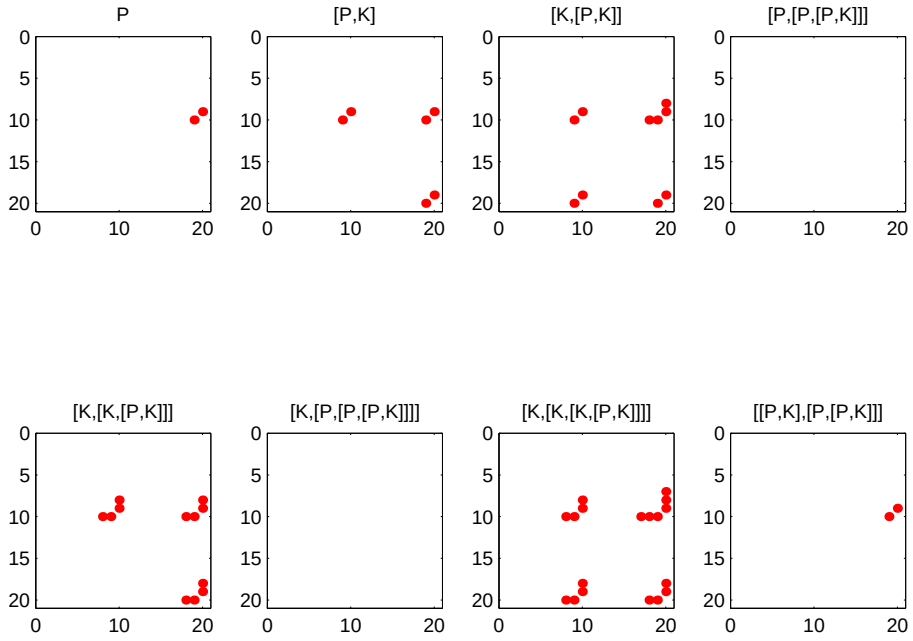


FIGURE 4.2. *The sparsity pattern in the generation of* $X(t)$.

It is trivial to observe that $[P,[P,[P,K]]] = O$. This is not just a consequence of the reduction to a butterfly form, but of a more general result.

PROPOSITION 4.1. *Let* A *be a* $2n \times 2n$ *matrix, partitioned in* $n \times n$ *blocks, and assume that* A *is of the form*

$$\begin{bmatrix} O & A_{1,2} \\ O & O \end{bmatrix}.$$

Then, for any matrix $C \in M_{2n,2n}$, *it is true that*

$$[A,[A,[A,C]]] = O.$$

Proof. Let us partition the matrix C in blocks of the same size of those of A,

$$C = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}.$$

By direct computation, we observe that

$$[A,C] = \begin{bmatrix} D_{1,1} & D_{1,2} \\ O & D_{2,2} \end{bmatrix},$$

where $D_{1,1} = A_{1,2}C_{2,1}$, $D_{1,2} = A_{1,2}C_{2,2} - C_{1,1}A_{1,2}$ and $D_{2,2} = -C_{2,1}A_{1,2}$. Similarly,

$$[A,[A,C]] = \begin{bmatrix} O & F_{1,2} \\ O & O \end{bmatrix},$$

where $F_{1,2} = -2A_{1,2}C_{1,2}A_{1,2}$. Finally, it is immediate to check that $[A,[A,[A,C]]] = O$. □

Note that the lemma is valid in the more general case, when A is $n \times n$, $A1,2$ is $n_1 \times n_2$ and $n_1 + n_2 = n$. The (trivial) extension of the proof is left to the reader.

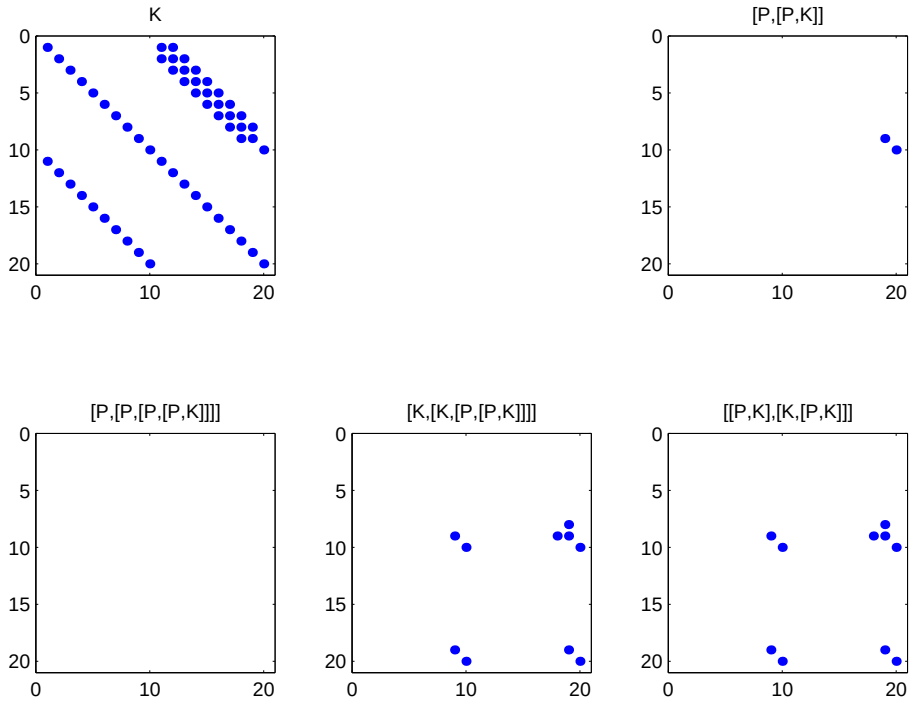


FIGURE 4.3. *Commutators in the expansion of* $Y(t)$.

The terms required for the generation of $Y(t)$ are displayed in Figure 4.3 and the superposition of Y_5 (darker shade) and X_5 (lighter shade) is displayed in Figure 4.4 (for convenience, we have plotted X_5 with $\times$)

In Figure 4.4 we observe that no fill-in is introduced at least up to order 6. This is most welcome news because we do not need to use extra computation to annihilate further entries.

4.3. The cost of reduction to a butterfly form. Assume that we have already partially reduced the matrix M to a butterfly from, where the blocks L_k, M_k, N_k have dimension k and N_k, M_k are symmetric (see figure (4.5)). To set to zero the terms in the leading column of N_k we apply symplectic Givens rotations from the left. Each of this rotations requires about $6k$ operations (multiplications and additions) for the update of L_k, M_k, and $3k$ operations for the update of N_k, for a total of $9k$ operations. When we apply their transpose from the right, we can take into account the symmetry of the $(1,2)$ and $(2,1)$ blocks, so that only L_k needs be updated, hence further $6k$ operations, for a total of $15k$ operations. A similar count holds for the

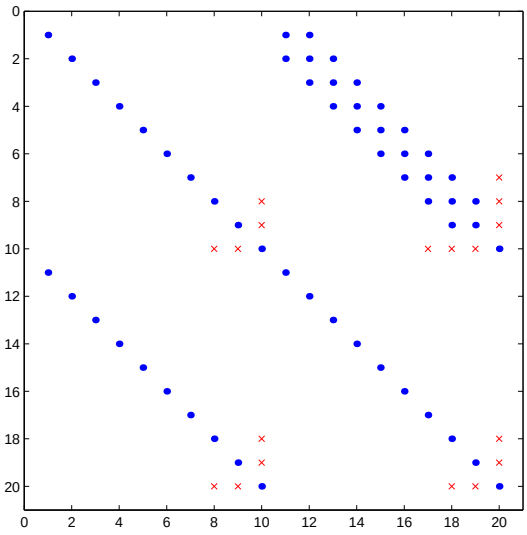


FIGURE 4.4. *Superposition of X_5 (crosses) and Y_5 (dots). No fill-in is introduced in the $\mathfrak{k}$ subalgebra.*

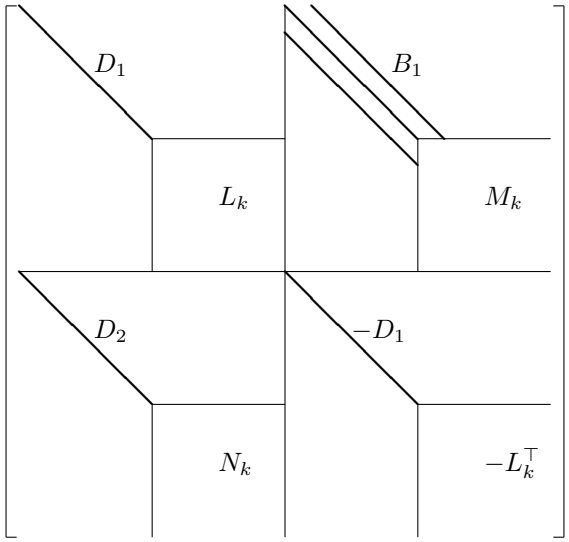


FIGURE 4.5. *A matrix in $\mathfrak{sp}(n)$ partially reduced to a butterfly form.*

Givens rotations that are applied to annihilate the top-row elements of L_k, hence *in toto* Givens rotations account for $30k$. Since for each column/row there are k of those Givens rotation, for matrices of decreasing dimension, neglecting lower order terms we have

$$\text{Total cost of Givens rotations} \approx 30 \sum_{k=1}^{n} k^2 \approx 10n^3.$$

The application of Householder symplectic reflections reduces to the application of standard Householder to the blocks L_k, M_k, N_k, and costs $3\frac{1}{3}n^3, n^3$ and n^3 respec-

tively, since the two latter blocks are symmetric. We need to apply two sets of such Householder reflections, for a total of

$$\text{Total cost of Householder reflections} \approx 10\frac{2}{3}n^3 \text{ operations.}$$

Since the cost of Gauss transformations is of a lower order of magnitude, the total cost of reduction to a butterfly form is

$$20\frac{2}{3}n^3 \text{ operations,}$$

which is not really prohibitive, given that the matrix has dimension $2n$.

5. A divide and conquer strategy. In what follows, we shall introduce an alternative approach, that can be particularly useful in the context of very large n and parallel computing. The main idea is to choose an automorphism Ad_S so that the matrix Y_p is reducible, hence, the exponential of each submatrix can be computed separately and possibly in parallel.

5.1. Skew-symmetric matrices. We again commence our exposition with $Z \in \mathfrak{so}(n)$ and assume that it is already in tridiagonal form. Our point of departure is to consider an inner automorphism Ad_G where

$$G = \begin{bmatrix} I_{n_1\times n_1} & O_{n_1\times n_2} \\ O_{n_2\times n_1} & -I_{n_2\times n_2} \end{bmatrix},$$

where $n_1 + n_2 = n$: an obvious choice is $n_1 = \lfloor n/2 \rfloor$, however, other choices are possible, e.g. the index corresponding to the least off-diagonal element. Then,

$$\mathfrak{k} \ni \begin{bmatrix} K^{(1)}_{n_1\times n_1} & O_{n_1\times n_2} \\ O_{n_2\times n_1} & K^{(2)}_{n_2\times n_2} \end{bmatrix}, \qquad \mathfrak{p} \ni \begin{bmatrix} O_{n_1\times n_1} & P^{(1)}_{n_1\times n_2} \\ P^{(2)}_{n_2\times n_1} & O_{n_2\times n_2} \end{bmatrix}.$$

Therefore,

$$K = \begin{bmatrix} K_1 & O \\ O & K_2 \end{bmatrix}, \qquad P = \begin{bmatrix} O & P_1 \\ P_2 & O \end{bmatrix}.$$

We stress that both K_1 and K_2 are tridiagonal while P_1 and P_2 have a single nonzero entry, in the lower left and upper right corner respectively. We write $P_1 = c_1\mathbf{e}_{n_1,n_1}\mathbf{e}^\top_{n_2,1}$, $P_2 = c_2\mathbf{e}_{n_2,1}\mathbf{e}^\top_{n_1,n_1}$, where $\mathbf{e}_{m,k} \in \mathbb{R}^m$ is the kth unit vector.

In Figures 5.1 and 5.2 we display the sparsity pattern of elements in $\mathfrak{p}$ and in $\mathfrak{k}$ respectively, while, in Figure 5.3, the matrices X and Y for different orders are superposed (truncations of X are denoted in lighter shade, while the darker shade corresponds to truncations of Y).

The following observations form the basis for an efficient divide-and-conquer algorithm to compute the exponential function in $\mathfrak{so}(n)$:

- All the commutators, hence also X and Y (up to the requisite order) can be evaluated in $\mathcal{O}(1)$ flops.
- The exact exponential of X reduces to that of a small matrix, hence can be evaluated in $\mathcal{O}(1)$ flops.
- Y is a reducible matrix.

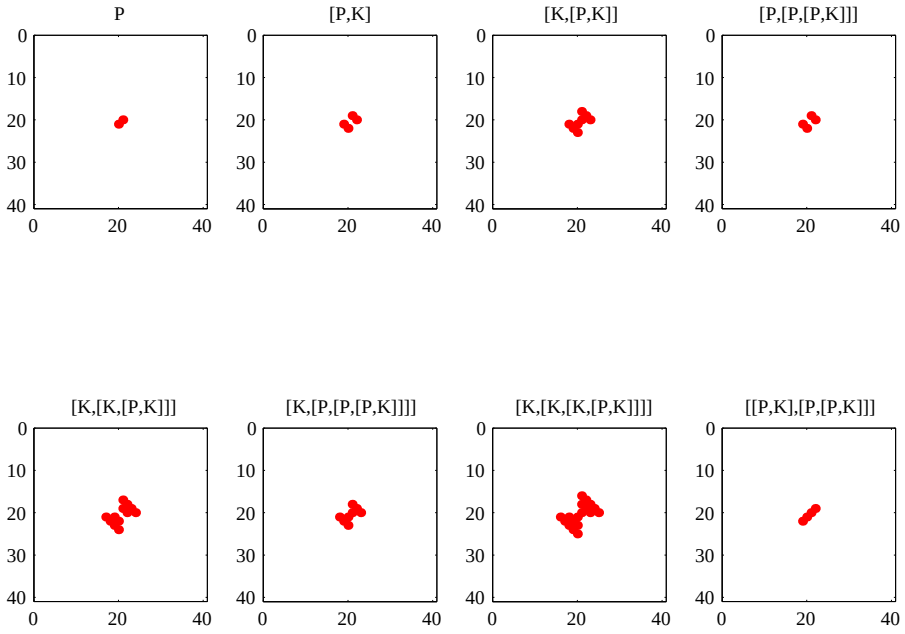


FIGURE 5.1. *Sparsity pattern of matrices in* $\mathfrak{p}$ *in the divide-and-conquer approach for skew-symmetric matrices.*

- The departure of Y from tridiagonal can be corrected in a small number of Givens rotations. Because of reducibility, we can act *separately* on each of the two components, hence the outcome is two tridiagonal matrices, of size $n_1 \times n_1$ and $n_2 \times n_2$, respectively. Again, the cost is $\mathcal{O}(1)$ flops.

We can now continue with the two pieces of Y in a similar vain, splitting them into progressively smaller pieces. All this is similar to many familiar techniques in numerical linear algebra, not least domain decomposition. Altogether, we require $\log_2 n$ stages to parcel out the exponential of a tridiagonal $Z \in \mathfrak{so}(n)$ into a product of rank-2 orthogonal matrices, although in practice this divide-and-conquer technique can terminate with matrices of higher rank.

5.2. General matrices. Let $Z \in \mathfrak{gl}(n)$ or $\mathfrak{sl}(n)$ and suppose that we have already brought it to an upper Hessenberg form. Proceeding as before, P_1 is a dense matrix, while $P_2 = c\mathbf{e}_{n_2,1}\mathbf{e}_{n_1,n_1}^\top$.

Again, we can always bring Y into an upper-Hessenberg form in $\mathcal{O}(1)$ Givens rotations. More interesting is the evaluation of $\exp X$. Note that

$$X = \begin{bmatrix} O & X_1 \\ X_2 & O \end{bmatrix},$$

where X_1 is $n_1 \times n_2$ and dense, while X_2 is $n_2 \times n_1$ and zero except for a $q \times q$ block in the upper right corner, where $q \geq 1$. Let

$$V = X_1 X_2, \qquad W = X_2 X_1.$$

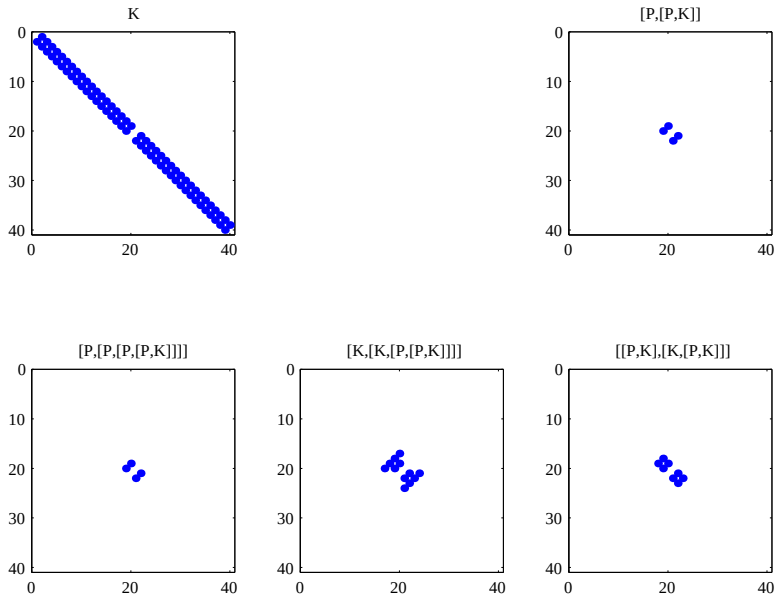


FIGURE 5.2. *Sparsity pattern of matrices in* $\mathfrak{k}$ *in the divide-and-conquer approach for skew-symmetric matrices.*

Then

$$X^{2m} = \begin{bmatrix} V^m & O \\ O & W^m \end{bmatrix}, \quad X^{2m+1} = \begin{bmatrix} V^m & O \\ O & W^m \end{bmatrix} \begin{bmatrix} O & X_1 \\ X_2 & O \end{bmatrix}, \qquad m \in \mathbb{Z}_+,$$

therefore

$$\exp(X) = \begin{bmatrix} C(V) & S(V)X_1 \\ S(W)X_2 & C(W) \end{bmatrix},$$

where

$$C(Z) = \sum_{m=0}^{\infty} \frac{Z^m}{(2m)!} = \cosh Z^{1/2}, \qquad S(Z) = \sum_{m=0}^{\infty} \frac{Z^m}{(2m+1)!} = \sinh Z^{1/2}.$$

We write X_1 and X_2 in a compound form,

$$X_1 = \begin{bmatrix} T_{1,1} & T_{1,2} \\ T_{2,1} & T_{2,2} \end{bmatrix}, \qquad X_2 = \begin{bmatrix} O & R \\ O & O \end{bmatrix},$$

where

$$T_{1,1} \in \mathrm{M}_{(n_1-p)\times p}, \quad T_{1,2} \in \mathrm{M}_{(n_1-p)\times(n_2-p)}, \quad T_{2,1} \in \mathrm{M}_{p\times p}, \quad T_{2,2} \in \mathrm{M}_{p\times(n_2-p)}$$

and $R \in \mathrm{M}_{p\times p}$, where $\mathrm{M}_{n\times m}$ denotes the set of $n \times m$ matrices. Therefore

$$V = \begin{bmatrix} O & V_1 \\ O & V_2 \end{bmatrix}, \qquad W = \begin{bmatrix} W_1 & W_2 \\ O & O \end{bmatrix},$$

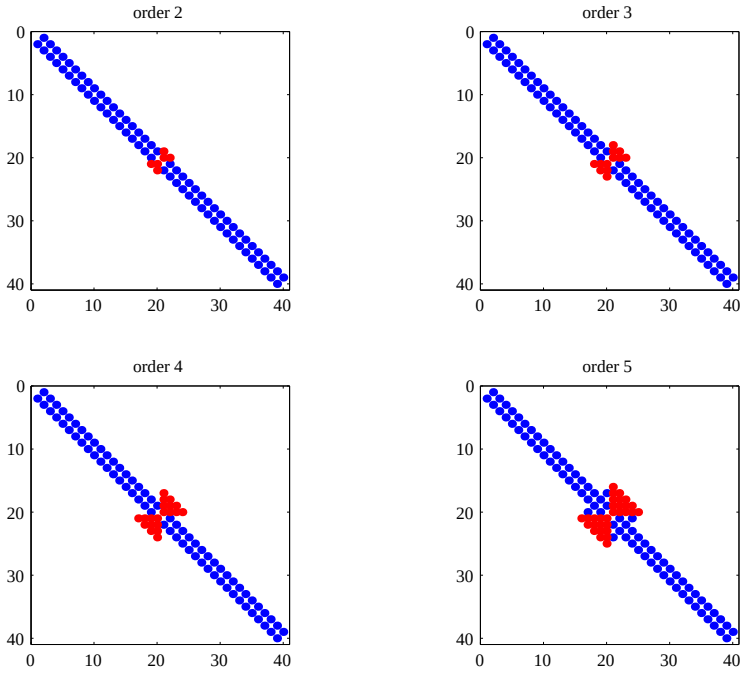


FIGURE 5.3. *Superposition of truncations of $X(t)$ (lighter shade) and $Y(t)$ (darker shade) of order $2,3,4,5$ in the divide-and-conquer approach for skew-symmetric matrices.*

where

$$V_1 = T_{1,1}R \in \mathrm{M}_{(n_1-p)\times p}, \quad V_2 = T_{2,1}R \in \mathrm{M}_{p\times p},$$
$$W_1 = RT_{2,1} \in \mathrm{M}_{p\times p}, \quad W_2 = RT_{2,2} \in \mathrm{M}_{p\times(n_2-p)}.$$

In particular, note that $V_2, W_1 \in \mathrm{M}_{p\times p}$, hence they are square and *small!*

We can easily prove by induction that

$$V^m = \begin{bmatrix} O & V_1V_2^{m-1} \\ O & V_2^m \end{bmatrix}, \qquad W^m = \begin{bmatrix} W_1^m & W_1^{m-1}W_2 \\ O & O \end{bmatrix}, \qquad m \in \mathbb{Z}.$$

Therefore simple calculation affirms that

$$C(V) = \begin{bmatrix} I & V_1V_2^{-1}[C(V_2)-I] \\ O & C(V_2) \end{bmatrix},$$
$$C(W) = \begin{bmatrix} C(W_1) & [C(W_1)-I]W_1^{-1}W_2 \\ O & I \end{bmatrix},$$
$$S(V)X_1 = \begin{bmatrix} I & V_1[S(V_2)-I]V_2^{-1} \\ O & S(V_2) \end{bmatrix} \begin{bmatrix} T_{1,1} & T_{1,2} \\ T_{2,1} & T_{2,2} \end{bmatrix},$$
$$S(W)X_2 = \begin{bmatrix} O & S(W_1)R \\ O & O \end{bmatrix}.$$

Given that $p \ll n_1, n_2$ and $n_1 + n_2 = n$, we have the following cost (disregarding lower-order terms),

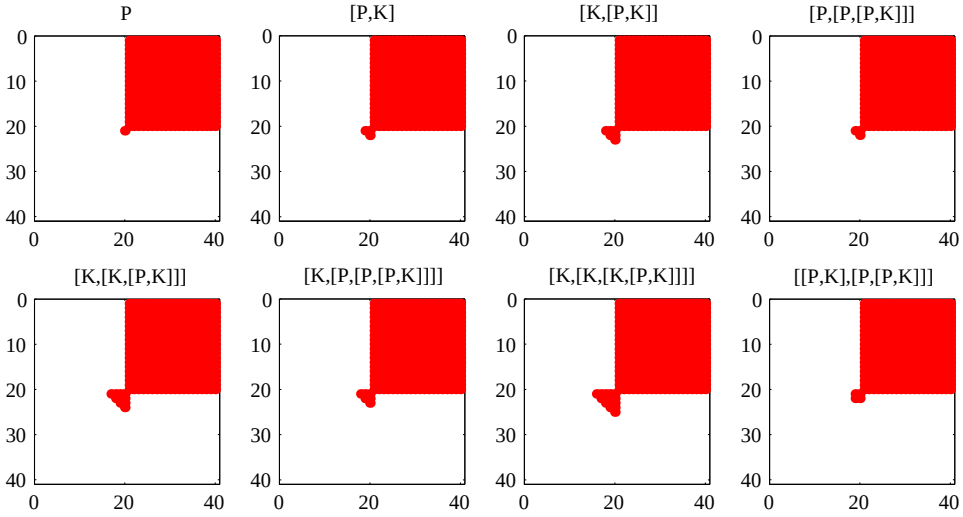


FIGURE 5.4. *Sparsity structure for the first terms in* $\mathfrak{p}$ *in the divide-and-conquer approach: upper Hessenberg matrices.*

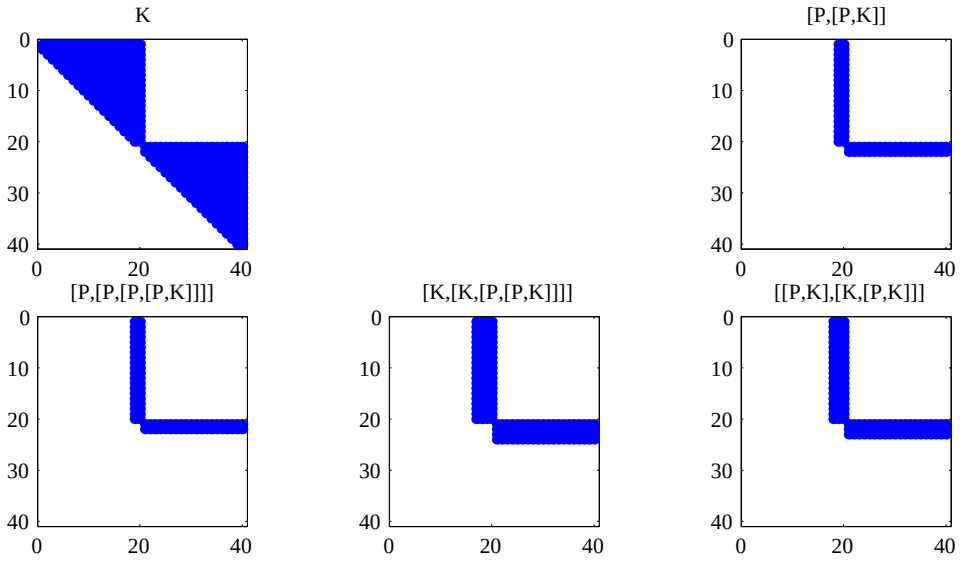


FIGURE 5.5. *Sparsity structure for the first terms in* $\mathfrak{k}$ *in the divide-and-conquer approach: upper Hessenberg matrices.*

1. Computing $C(V)$: $n_1 p^2$ flops;
2. Computing $C(W)$: $n_2 p^2$ flops;
3. Computing $S(V)$: $n_1 p^2$ flops;
4. Multiplying $S(V)X_1$: $n_1 n_2 p + n p^2$;

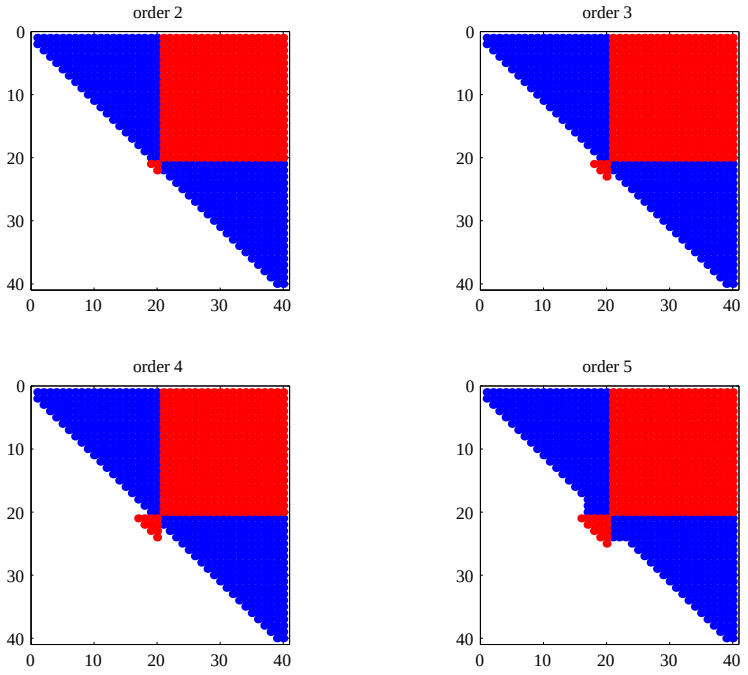


FIGURE 5.6. *A superposition of X_5 and Y_5 in the divide-and-conquer approach (upper Hessenberg matrices).*

5. Computing $S(W)X_2$: p^3 flops.[1]

Hence, altogether the cost is $n_1 n_2 p$: if $n_1 = n_2 = n/2$ then the entire cost of computing the exponential *exactly* is just $\frac{1}{4}n^2 p$.

Suppose that $n = 2^s$ and $n_1 = n_2 = 2^{s-1}$, whence the cost is $\approx p2^{2s-2}$. Moreover, we continue with the divide-and-conquer technique. In the next stage, we have two $2^{s-1} \times 2^{s-1}$ matrices, then four $2^{s-2} \times 2^{s-2}$ matrices and so on. The entire cost of computing all the exponentials then becomes

$$p\sum_{r=1}^{s} 2^{2s-r-1} = \tfrac{1}{2}pn(n-1) \approx \tfrac{1}{2}pn^2.$$

The above is true on a serial machine. If the calculations for different pieces of the matrix are performed in parallel, we have instead just a single $2^{s+1-r} \times 2^{s+1-r}$ matrix to deal with in the rth stage and the overall cost is

$$p\sum_{r=1}^{s} 2^{s+1-r} \approx \tfrac{1}{3}pn^2.$$

5.3. The cost of computing commutators. Assume that P and K are as in figures (5.4) and (5.5),

$$P = \begin{bmatrix} O & P_1 \\ P_2 & O \end{bmatrix}, \qquad K = \begin{bmatrix} K_1 & O \\ O & K_2 \end{bmatrix}, \tag{5.1}$$

[1]Note that we do not need to compute $S(W)$ first, just $S(W_1)$.

where the blocks P_i and K_i have dimension k. Note that

$$\begin{aligned}\left[\begin{bmatrix} O & P_1 \\ P_2 & O \end{bmatrix}, \begin{bmatrix} O & R_1 \\ R_2 & O \end{bmatrix}\right] &= \begin{bmatrix} O & P_1R_2 - R_1P_2 \\ P_2R_1 - R_2P_1 & O \end{bmatrix}, \\ \left[\begin{bmatrix} O & P_1 \\ P_2 & O \end{bmatrix}, \begin{bmatrix} K_1 & O \\ O & K_2 \end{bmatrix}\right] &= \begin{bmatrix} O & P_1K_2 - K_1P_1 \\ P_2K_1 - K_2P_1 & O \end{bmatrix}.\end{aligned} \tag{5.2}$$

¿From (5.2) we conclude that the most expensive commutators are those of the form $[P, K]$, for which we need to compute $P_1K_2 - K_1P_1$, amounting to about $2k^3$ operations (counting addition and multiplication). All the remaining commutators are of lower complexity, which we ignore. For a splitting of order 5 (as depicted in Figure 5.6), we need to compute seven such commutators, amounting to $14k^3$. Next, assume that $k = 2^s$ and that there are $2^{\log_2 n - s}$ such blocks: for order five, the total cost of commutators (disregarding lower order terms) amounts to

$$14n \sum_{s=1}^{\log_2 n} 2^{2s} \approx 15\frac{1}{3}n^3$$

operations on a serial machine. When implementing this in parallel on $\log_2 n$ processors (that is, when the commutators of blocks of dimension 2^s are evaluated simultaneously), the cost reduces to

$$14 \sum_{s=1}^{\log_2 n} 2^{3s} \approx 2n^3.$$

6. Other groups. In this section we discuss briefly generalized polar decompositions for a number of more unusual Lie groups which, nonetheless, feature in applications.

6.1. Lorentz-type groups $\mathrm{SO}(p, q)$**.** Let

$$J = \begin{pmatrix} I_p & O \\ O & -I_q \end{pmatrix},$$

and consider the group $\mathrm{SO}(p, q) = \{x : xJx^\top = J\}$. As is well known, the corresponding algebra is $\mathfrak{so}(p, q) = \{Z : ZJ = -JZ^\top\}$. The block form of Z is

$$Z = \begin{pmatrix} Z_1 & Z_2 \\ Z_2^\top & Z_3 \end{pmatrix},$$

where Z_1 and Z_3 are skew-symmetric. The most widespread Lorenz-type groups in applications are $\mathrm{SO}(3, 1)$ and $\mathrm{SO}(5, 2)$, for which it is not too costly to compute the exponential exactly given the low dimension. Algorithms for computing the exact exponential of these matrices have been proposed in (Leite & Crouch 1999).

In what follows, we focus instead on the less ordinary case when $p + q = n$ is large, yet $p \ll q$. The basic idea consists of splitting

$$Z = \begin{pmatrix} O & Z_2 \\ Z_2^\top & O \end{pmatrix} + \begin{pmatrix} Z_1 & O \\ O^\top & Z_3 \end{pmatrix} = P + K$$

so that the problem is reduced to computing the exponential of skew-symmetric matrices and that of a (symmetric) bordered matrix.

The only issue that can cause complications is the computation of commutators with P, K (especially if we desire high order). To do this in a cheaper manner, we consider the matrix

$$H = \begin{pmatrix} I & O \\ O^\top & H_2 \end{pmatrix}$$

where H_2 is the matrix that QR factorizes P (a product of p elementary Householder reflections). Computing commutators with P (ad_P) costs now $2p^2q$. Commutators with K are more expensive—at present we do not see how this can be avoided but perhaps one can exploit skew symmetry.

However, these order conditions are used only once. Once we have split into the symmetric and skew-symmetric parts, the problem reduces to the approximation of exponentials of skew-symmetric matrices, which has been described at length earlier in this paper.

6.2. Isotropy groups. In this section we consider the computation of the exponential in isotropy groups. Recall that the (left) isotropy group G_V at $V \in \mathrm{M}_{n\times m}$ is the group of matrices x that leaves V fixed under left multiplication,

$$G_V = \{x \in \mathrm{GL}(n) : xV = V\}$$

(see for instance (Olver 1993)). The corresponding algebra can be easily computed,

$$\mathfrak{g}_V = \{X \in \mathfrak{gl}(n) : XV = V\}.$$

Let us assume that $m \le n$ and that V has rank m (if V has rank less then m then it is possible to ignore some of its columns so that the resulting matrix has full rank). In that case, the problem essentially reduces to the isotropy group (isomorphic to G_V)

$$G_{\tilde R} = \{y : y\tilde R = \tilde R\}, \qquad \tilde R = \begin{bmatrix} R \\ O \end{bmatrix}$$

where R is $m \times m$ upper triangular and $G_{\tilde R} = Q^\top G_V Q$, Q being the orthogonal matrix that performs the QR factorization of V, i.e. $V = Q\tilde R$. In what follows, we abuse notation and write G_R instead of $G_{\tilde R}$, hoping that this does not cause confusion of types.

Let us study in greater detail the elements of G_R. Assume that

$$y = \begin{bmatrix} y_{1,1} & y_{1,2} \\ y_{2,1} & y_{2,2} \end{bmatrix},$$

where $y_{1,1}$ is $m \times m$, $y_{1,2}$ and $y_{2,1}^\top$ are $m \times (n-m)$ and $y_{2,2}$ is $(n-m) \times (n-m)$. Imposing $y\tilde R = \tilde R$ we obtain the conditions:

$$\begin{aligned} y_{1,1}R = R \quad &\Rightarrow \quad y_{1,1} = I_{m\times m} \\ y_{2,1}R = O_{(n-m)\times m} \quad &\Rightarrow \quad y_{2,1} = O_{(n-m)\times m} \\ y_{1,2}, y_{2,2} \quad &\text{arbitrary} \end{aligned}$$

(recall that R has full rank, hence it is invertible).

Correspondingly, at the algebra level, we set

$$Y = \begin{bmatrix} Y_{1,1} & Y_{1,2} \\ Y_{2,1} & Y_{2,2} \end{bmatrix},$$

which, in tandem with the algebra conditions, implies

$$\begin{aligned} Y_{1,1} &= O_{m\times m}, \\ Y_{2,1} &= O_{(n-m)\times m}, \\ Y_{1,2}, Y_{2,2} &\quad \text{arbitrary.} \end{aligned}$$

Note that $Y_{1,2}, Y_{2,2}$ can be considered as free parameters: although their action does not change $\tilde{R}$, they do move points around $\tilde{R}$. As a possible application, in (Lewis & Olver 2002) isotropy at a point is used to improve retention of qualitative features by numerical integrators for ordinary differential equations.

The exponential of Y can be evaluated exactly by the formula

$$\exp(tY) = \begin{bmatrix} I_{m\times m} & Y_{1,2}\phi(Y_{2,2}) \\ O & \exp(tY_{2,2}) \end{bmatrix},$$

where

$$\phi(Z) = Z^{-1}[\exp(Z) - I] = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} Z^k.$$

Thus, the problem reduces to computing the exponential of $Y_{2,2}$, which is of dimension $(n-m)$. Now, $Y_{2,2}$ can be reduced to Hessenberg form and its exponential computed as in the general $\mathrm{GL}(n)$ case.

6.3. Scaling groups. We commence as in the case of the isotropy groups. A one-parameter curve in the scaling group G_V^{sc} of $V = (\mathbf{v}_1, \ldots, \mathbf{v}_m) \in \mathbb{R}^{n\times m}$,

$$G_V^{\mathrm{sc}} = \{x : \exists \lambda_1, \lambda_2, \ldots, \lambda_m \text{ such that } x\mathbf{v}_i = \lambda_i \mathbf{v}_i, i = 1, \ldots, m\},$$

satisfies

$$x(t)V = V\Lambda(t),$$

where $\Lambda(t)$ is a smooth diagonal matrix function (Olver 1993). Again, performing a QR decomposition of V, we find that G_V^{sc} is isomorphic to $Q^{\top} G_V^{\mathrm{sc}} Q = G_{\tilde{R}}^{\mathrm{sc}}$, the scaling group of the upper triangular matrix $\tilde{R}$, where $Q\tilde{R} = V$.

At the algebra level $\mathfrak{g}_{\tilde{R}}^{\mathrm{sc}}$, we set

$$Y = \begin{bmatrix} Y_{1,1} & Y_{1,2} \\ Y_{2,1} & Y_{2,2} \end{bmatrix},$$

which, in tandem with the algebra condition

$$Y\tilde{R} = \tilde{R}\Lambda'(t)$$

implies that

$$\begin{aligned} Y_{1,1} &= R\Lambda'(t)R^{-1}, \\ Y_{2,1} &= O_{(n-m)\times m}, \\ Y_{1,2}, Y_{2,2} &\quad \text{arbitrary.} \end{aligned}$$

Again, $Y_{1,2}, Y_{2,2}$ can be considered as free parameters: their action does not move/scale $\tilde{R}$ but only the points in its neighbourhood. It is also useful to note that $Y_{1,1}$ is upper triangular.

Thus, the problem reduces to computing exponentials of block-upper triangular matrices of the form

$$Y = \begin{bmatrix} Y_{1,1} & Y_{1,2} \\ O & Y_{2,2} \end{bmatrix},$$

where $Y_{1,1}$ is upper triangular.

In general, the exponential of a block triangular matrix can be evaluated exactly by the formula

$$\exp\left(t \begin{bmatrix} A & B \\ O & C \end{bmatrix}\right) = \begin{bmatrix} \exp(tA) & \int_0^t \mathrm{e}^{(t-\tau)A} B \mathrm{e}^{\tau C} \mathrm{d}\tau \\ 0 & \exp(tC) \end{bmatrix}, \tag{6.1}$$

however, the integral might be a difficult to compute exactly. It could be approximated by quadrature formulæ, but this will require the computation of roots of matrices, adding an extra layer of complexity.

We could again use the *divide and conquer* approach: split

$$Y = K + P, \qquad K = \begin{bmatrix} Y_{1,1} & O \\ O & Y_{2,2} \end{bmatrix}, \quad P = \begin{bmatrix} O & Y_{2,1} \\ O & O \end{bmatrix}.$$

It can be observed that Y is block upper triangular and so are K and P. Since triangular (and block triangular) matrices form subalgebras, $\mathfrak{k}$ and $\mathfrak{p}$ also consist of block triangular matrices (in other words, the lower (2,1) block never fills in the $\mathfrak{p}$-part).

Now,

$$[P, K] = \begin{bmatrix} O & Y_{1,2}Y_{2,2} - Y_{1,1}Y_{1,2} \\ O & O \end{bmatrix},$$

moreover, if $P_1, P_2 \in \mathfrak{p}$, it is easily verified that

$$O = [P_1, P_2] \in \mathfrak{k}.$$

In other words, K *does not need any order corrections* and the only nonzero commutators are those of the form

$$[K, [K, [\cdots, [K, [P, K]]]]]$$

in the $\mathfrak{p}$ part.

Moreover, observe that

$$\begin{bmatrix} O & B \\ O & O \end{bmatrix}^2 = O,$$

therefore

$$\exp(t\tilde{P}) = I + t\tilde{P}, \qquad \tilde{P} \in \mathfrak{p},$$

while, for matrix $\tilde{K} = \begin{bmatrix} A & O \\ O & C \end{bmatrix} \in \mathfrak{k}$, one has

$$\exp(t\tilde{K}) = \begin{bmatrix} \exp(tA) & O \\ O & \exp(tC) \end{bmatrix}.$$

An alternative is to expand the integral in (6.1) in Taylor series and truncate to appropriate order.

Acknowledgments. This work was completed while both authors were visiting the Centre for Advanced Studies, Oslo, Norway. They both wish to thank the staff and the members for making their stay so pleasant and productive. Thanks also to Brad Baxter, Stein Krogstad, Hans Z. Munthe-Kaas and Beresford Parlett for useful discussion and suggestions.

REFERENCES

Benner, P., Faßbender, H. & Watkins, D. S. (1999), 'SR and SZ algorithms for the symplectic (butterfly) eigenproblem', *Lin. Alg. Appl.* **287**, 41–76.

Celledoni, E. & Iserles, A. (2000), 'Approximating the exponential from a Lie algebra to a Lie group', *Math. Comp.* **69**, 1457–1480.

Celledoni, E. & Iserles, A. (2001), 'Methods for the approximation of the matrix exponential in a Lie-algebraic setting', *IMA J. Num. Anal.* **21**, 463–488.

Faßbender, H. (2000), 'The parametrixed SR algorithm for symplectic (butterfly) matrices', *Math. Comp.* **70**(236), 1515–1541.

Golub, G. H. & van Loan, C. F. (1989), *Matrix Computations*, 2nd edn, John Hopkins, Baltimore.

Helgason, S. (1978), *Differential Geometry, Lie Groups and Symmetric Spaces*, Academic Press.

Hochbruck, M., Lubich, C. & Selhofer, H. (1998), 'Exponential integrators for large systems of differential equations', *SIAM J. Sci. Comput.* **19**(5), 1552–1574.

Horn, R. A. & Johnson, C. R. (1985), *Matrix Analysis*, Cambridge University Press.

Iserles, A., Munthe-Kaas, H., Nørsett, S. P. & Zanna, A. (2000), 'Lie-group methods', *Acta Numerica* **9**, 215–365.

Kang, F. & Shang, Z.-J. (1995), 'Volume-preserving algorithms or source-free dynamical systems', *Numer. Math.* **71**, 451–463.

Lawson, J. D. (1994), 'Polar and Ol'shanskii decompositions', *J. Reine Angew. Math.* **448**, 191–219.

Leite, F. S. & Crouch, P. (1999), 'Closed forms for the exponential mapping on matrix Lie groups based on Putzer's method', *J. of Math. Phys.* **40**(7), 3561–3568.

Lewis, D. & Olver, P. J. (2002), 'Geometric integration algorithms on homogeneous manifolds', *Found. Comp. Math.* To appear.

Moler, C. B. & van Loan, C. F. (1978), 'Nineteen dubious ways to compute the exponential of a matrix', *SIAM Rev.* pp. 801–836.

Munthe-Kaas, H., Quispel, R. G. W. & Zanna, A. (2001), 'Generalized polar decompositions on Lie groups with involutive automorphisms', *Found. Comp. Math.* **1**(3), 297–324.

Olver, P. J. (1993), *Applications of Lie Groups to Differential Equations*, GTM 107, Second edn, Springer-Verlag.

Saad, Y. (1992), 'Analysis of some Krylov subspace approximations to the matrix exponential operator', *SIAM J. Numer. Anal.* **29**, 209–228.

Zanna, A. (2000), Recurrence relation for the factors in the polar decomposition on Lie groups, Technical Report Report no. 192, Department of Informatics, University of Bergen, Norway. To appear in *Math. Comp.*

Zanna, A. & Munthe-Kaas, H. Z. (2002), 'Generalized polar decompositions for the approximation of the matrix exponential', *SIAM J. Matrix Anal.* **23**(3), 840–862.

Appendix A. Herewith, we report the coefficients for the commutators in (3.16).

$$
\begin{aligned}
g_{n,n-3} &= -\beta_{n-3} c_{n,n-2} \\
g_{n,n-2} &= (\alpha_n - \alpha_{n-2}) c_{n,n-2} - \beta_{n-2} c_{n,n-1} \\
g_{n,n-1} &= (\alpha_n - \alpha_{n-1}) c_{n-1} - \gamma_{n-2} c_{n,n-2} \\
g_{n-3,n} &= \gamma_{n-3} c_{n-2,n} \\
g_{n-2,n} &= -(\alpha_n - \alpha_{n-2}) c_{n-2,n} + \gamma_{n-2} c_{n-1,n} \\
g_{n-1,n} &= -(\alpha_n - \alpha_{n-1}) c_{n-1,n} - \beta_{n-3} c_{n-2,n}
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
h_{n,n-4} &= -\beta_{n-4} g_{n,n-3} \\
h_{n,n-3} &= (\alpha_n - \alpha_{n-3}) g_{n,n-3} - \beta_{n-3} g_{n,n-2} \\
h_{n,n-2} &= (\alpha_n - \alpha_{n-2}) g_{n,n-2} - \gamma_{n-3} g_{n,n-3} - \beta_{n-2} g_{n,n-1} \\
h_{n,n-1} &= (\alpha_n - \alpha_{n-1}) g_{n,n-1} - \gamma_{n-2} g_{n,n-2} \\
h_{n-4,n} &= \gamma_{n-4} g_{n-3,n} \\
h_{n-3,n} &= -(\alpha_n - \alpha_{n-3}) g_{n-3,n} + \gamma_{n-3} g_{n-2,n} \\
h_{n-2,n} &= -(\alpha_n - \alpha_{n-2}) g_{n-2,n} + \beta_{n-3} g_{n-3,n} + \gamma_{n-2} g_{n-1,n} \\
h_{n-1,n} &= -(\alpha_n - \alpha_{n-1}) g_{n-1,n} + \beta_{n-2} g_{n-2,n}
\end{aligned} \tag{A.2}
$$

$$
\begin{aligned}
j_{n,n-5} &= \beta_{n-5} h_{n,n-4} \\
j_{n,n-4} &= (\alpha_n - \alpha_{n-4}) h_{n,n-4} - \beta_{n-4} h_{n,n-2} \\
j_{n,n-3} &= -\gamma_{n-4} h_{n,n-4} + (\alpha_n - \alpha_{n-3}) h_{n,n-3} - \beta_{n-3} h_{n,n-1} \\
j_{n,n-2} &= -\gamma_{n-3} h_{n,n-4} + (\alpha_n - \alpha_{n-2}) h_{n,n-2} - \beta_{n,2} h_{n,n-1} \\
j_{n,n-1} &= -\gamma_{n-2} h_{n,n-2} + (\alpha_n - \alpha_{n-1}) h_{n,n-1} \\
j_{n-5,n} &= \gamma_{n-5} h_{n-3,n} \\
j_{n-4,n} &= -(\alpha_n - \alpha_{n-4}) h_{n-4,n} + \gamma_{n-4} h_{n-3,n} \\
j_{n-3,n} &= \beta_{n-4} h_{n-4,n} - (\alpha_n - \alpha_{n-3}) h_{n-4,n} + \gamma_{n-3} h_{n-2,n} \\
j_{n-2,n} &= \beta_{n-3} h_{n-3,n} - (\alpha_n - \alpha_{n-2}) h_{n-2,n} + \gamma_{n-2} h_{n-1,n} \\
j_{n-1,n} &= \beta_{n-2} h_{n-2,n} - (\alpha_n - \alpha_{n-1}) h_{n-1,n}
\end{aligned} \tag{A.3}
$$

$$
\begin{aligned}
k_{n,n-2} &= c_{n,n-1} d_{n-1,n-2} - c_{n,n-2} d_{n,n} \\
k_{n,n-1} &= c_{n,n-2} d_{n-2,n-1} - c_{n,n-1} (d_{n,n} - d_{n-1,n-1}) \\
k_{n-2,n} &= c_{n-2,n} d_{n,n} - d_{n-2,n-1} c_{n-1,n} \\
k_{n-1,n} &= (d_{n,n} - d_{n-1,n-1}) c_{n-1,n} - d_{n-1,n-2} c_{n-2,n}
\end{aligned} \tag{A.4}
$$

$$
\begin{aligned}
l_{n-1,n-4} &= \beta_{n-3}\beta_{n-4}d_{n-1,n-2}\\
l_{n-1,n-3} &= -\beta_{n-3}[(2\alpha_{n-1}-\alpha_{n-2}-\alpha_{n-3})d_{n-1,n-2}-d_{n-1,n-1}\beta_{n-2}]\\
l_{n-1,n-2} &= [(\alpha_{n-1}-\alpha_{n-2})^2+2\beta_{n-2}\gamma_{n-2}+\beta_{n-3}\gamma_{n-3}]d_{n-1,n-2}\\
&\quad -(\alpha_{n-1}-\alpha_{n-2})\beta_{n-2}d_{n-1,n-1}-2\beta_{n-1}^2d_{n-2,n-1}\\
l_{n-1,n-1} &= -l_{n-2,n-2}\\
l_{n-2,n-3} &= \beta_{n-3}(-2\gamma_{n-2}d_{n-1,n-2}+\beta_{n-2}d_{n-2,n-1})\\
l_{n-2,n-2} &= (\alpha_{n-1}-\alpha_{n-2})(\gamma_{n-2}d_{n-1,n-2}+\beta_{n-2}d_{n-2,n-1})\\
&\quad -2\gamma_{n-2}\beta_{n-2}d_{n-1,n-1}\\
l_{n-2,n-1} &= [(\alpha_{n-1}-\alpha_{n-2})^2+2\beta_{n-2}\gamma_{n-2}+\gamma_{n-3}\beta_{n-3}]d_{n-2,n-1}\\
&\quad -(\alpha_{n-1}-\alpha_{n-2})\gamma_{n-2}d_{n-1,n-1}\\
l_{n-3,n-2} &= \gamma_{n-3}(\gamma_{n-2}d_{n-1,n-2}-2\beta_{n-2}d_{n-2,n-1})\\
l_{n-3,n-1} &= -\gamma_{n-3}[(2\alpha_{n-1}-\alpha_{n-2}-\alpha_{n-3})d_{n-2,n-1}-\gamma_{n-2}d_{n-1,n-1}]\\
l_{n-4,n-1} &= \gamma_{n-3}\gamma_{n-4}d_{n-2,n-1}
\end{aligned}
\tag{A.5}
$$

$$
\begin{aligned}
m_{n-1,n-3} &= c_{n-1,n}g_{n,n-3}\\
m_{n-1,n-2} &= c_{n-1,n}g_{n,n-2}-c_{n,n-2}g_{n-1,n}\\
m_{n-1,n-1} &= c_{n-1,n}g_{n,n-1}-c_{n,n-1}g_{n-1,n}\\
m_{n-2,n-3} &= c_{n-2,n}g_{n,n-3}\\
m_{n-2,n-2} &= c_{n-2,n}g_{n,n-2}-c_{n,n-2}g_{n-2,n}\\
m_{n-2,n-3} &= c_{n-2,n}g_{n,n-1}-c_{n,n-1}g_{n-2,n}\\
m_{n-3,n-2} &= -c_{n,n-2}g_{n-3,n}\\
m_{n-3,n-1} &= -c_{n,n-1}g_{n-3,n}\\
m_{n,n} &= -(m_{n-2,n-2}+m_{n-1,n-1}).
\end{aligned}
\tag{A.6}
$$