



HAL
open science

A Lanczos-like method for non autonomous linear ordinary differential equations

P.-L Giscard, Stefano Pozza

► **To cite this version:**

P.-L Giscard, Stefano Pozza. A Lanczos-like method for non autonomous linear ordinary differential equations. 2022. hal-03597313

HAL Id: hal-03597313

<https://hal.science/hal-03597313>

Preprint submitted on 4 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Lanczos-like method for non autonomous linear ordinary differential equations

Pierre-Louis Giscard^{1†} and Stefano Pozza^{2,3*†}

¹Université du Littoral Côte d'Opale, EA2597-LMPA-Laboratoire de Mathématiques Pures et Appliquées Joseph Liouville, Calais, France.

^{2*}Charles University, Faculty of Mathematics and Physics, Prague 8, Czech Republic.

³Associated member of ISTI, CNR, Pisa, Italy, and member of INdAM-GNCS group, Italy.

*Corresponding author(s). E-mail(s): pozza@karlin.mff.cuni.cz;
Contributing authors: giscard@univ-littoral.fr;

†These authors contributed equally to this work.

Abstract

The time-ordered exponential is defined as the function that solves a system of coupled first-order linear differential equations with generally non-constant coefficients. In spite of being at the heart of much system dynamics, control theory, and model reduction problems, the time-ordered exponential function remains elusively difficult to evaluate. The ***-Lanczos algorithm is a (symbolic) algorithm capable of evaluating it by producing a tridiagonalization of the original differential system. In this paper, we explain how the ***-Lanczos algorithm is built from a generalization of Krylov subspaces, and we prove crucial properties, such as the *matching moment property*. A strategy for its numerical implementation is also outlined and will be subject of future investigation.

Keywords: Lanczos algorithm, Matrix differential equations, Time-ordered exponential, Matching moments, Tridiagonal matrices, Ordinary differential equations

1 Introduction

Let $t' \geq t \in I \subseteq \mathbb{R}$ be variables—called times for convenience—in an interval I , and $\mathbf{A}(t')$ be an $N \times N$ time-dependent matrix. For a fixed t (usually $t = 0$), the time-ordered exponential of $\mathbf{A}(t')$ is defined as the unique solution $\mathbf{U}(t', t)$ of the non autonomous system of linear ordinary differential equations

$$\mathbf{A}(t')\mathbf{U}(t', t) = \frac{d}{dt'}\mathbf{U}(t', t), \quad \mathbf{U}(t, t) = \text{Id}, \quad t' \geq t, \quad (1.1)$$

with Id the identity matrix. Note that t represents the time on which the initial condition is given, and that the (unusual) notation $\mathbf{U}(t', t)$ will be useful later. If the matrix \mathbf{A} commutes with itself at all times, i.e., $\mathbf{A}(\tau_1)\mathbf{A}(\tau_2) - \mathbf{A}(\tau_2)\mathbf{A}(\tau_1) = 0$ for all $\tau_1, \tau_2 \in I$, then the time-ordered exponential is given by the matrix exponential $\mathbf{U}(t', t) = \exp\left(\int_t^{t'} \mathbf{A}(\tau) d\tau\right)$. However, when \mathbf{A} does not commute with itself at all times, the time-ordered exponential has no known explicit form in terms of \mathbf{A} and is rather denoted

$$\mathbf{U}(t', t) = \mathcal{T} \exp\left(\int_t^{t'} \mathbf{A}(\tau) d\tau\right),$$

with \mathcal{T} the time-ordering operator [1]. This expression, introduced by Dyson in 1952, is more a notation than an explicit form as the action of the time-ordering operator is very difficult to evaluate. In particular, $\mathbf{U}(t', t)$ does not have a Cauchy integral representation, and it cannot be evaluated via ordinary diagonalization. It is unlikely that a closed form expression for $\mathbf{U}(t', t)$ in terms of \mathbf{A} exists at all since even when \mathbf{A} is 2×2 , \mathbf{U} can involve very complicated special functions [2, 3].

Evaluating time-ordered exponentials is a central question in the field of system dynamics, in particular in quantum physics where \mathbf{A} is the Hamiltonian operator. Situations where this operator does not commute with itself are routinely encountered [4], and the departure of the time-ordered exponential from a straightforward matrix exponential is responsible for many peculiar physical effects [5–7]. Further applications are found via differential Lyapunov and Riccati matrix equations, which frequently appear in control theory, filter design, and model reduction problems [8–12]. Indeed, the solutions of such differential equations involve time-ordered exponentials [13–16].

In [17], we introduced a tridiagonal form for the matrix $\mathbf{A}(t')$ from which it is possible to express a time-ordered exponential via path-sum continued fractions of finite depth. More precisely, the described procedure formulates each element of a time-ordered exponential in terms of a finite and treatable number of scalar integro-differential equations. Such a tridiagonal form is obtained by using the so-called **-Lanczos algorithm*. The algorithm also appeared in [18] as the paper’s motivation. Despite being at the core of results in [17] and being the motivation of [18], the **-Lanczos algorithm* construction and the proofs

of related main properties have yet not appeared in a scientific journal. The present paper aims to solve such a gap in the literature as it constructs the $*$ -Lanczos algorithm from a (generalized) Krylov subspace perspective, proves the related *Matching Moment Property*, introduces a bound for the approximation error, and adds further results on the breakdown issue that may affect the method.

1.1 Existing analytical approaches: Pitfalls and drawbacks

In spite of the paramount importance of the time-ordered exponential, it is usually omitted from the literature on matrix functions. Until 2015, only two families of analytical approaches existed (numerical methods will be discussed in Section 5). The first one to have been devised relies on Floquet theory and necessitates $A(t')$ to be periodic (see, e.g., [4]). This method transforms Eq. (1.1) into an *infinite* system of coupled linear differential equations with *constant* coefficients. This system is then solved perturbatively at very low order, as orders higher than 2 or 3 are typically too involved to be treated. The second method was developed in 1954 by Wilhelm Magnus [19]. It produces an infinite series of nested commutators of A with itself at different times, the ordinary matrix exponential of which provides the desired solution $U(t', t)$. Magnus series are very much in use nowadays [4], especially because they guarantee that the approximation to $U(t', t)$ is unitary in quantum mechanical calculations [4]. Nevertheless, the Magnus series for $U(t', t)$ has a small (even if not so restrictively) convergence domain; see [20] and also [21–25].

In 2015, P.-L. G. et al. proposed a third method to obtain time-ordered exponentials using graph theory and necessitating only the entries $A(t')_{k\ell}$ to be bounded functions of time [26]. The method formulates any desired entry or group of entries of $U(t', t)$ as a branched continued fraction of *finite* depth and breadth. It has been successfully used to solve challenging quantum dynamic problems, see e.g. [27, 28]. This approach is unconditionally convergent and it provides exact expressions in terms of a finite number of integrals and Volterra equations. However, it suffers from a complexity drawback. Indeed, it requires one to find all the simple cycles and simple paths of a certain graph G . These are the walks on G which are not self-intersecting. Unfortunately, the problem of enumerating such walks is $\#P$ -complete [29], hindering the determination of exact solutions in large systems that must be treated using a further property of analytical path-sums called scale-invariance [28]. The present work with the results in [17, 18] solve this issue with a numerical outlook, by effectively mapping the dynamical graph G on a structurally simpler graph with well-chosen time-dependent edge weights. On this graph, the path-sum solution takes the form of an ordinary, finite, continued fraction.

1.2 The non-Hermitian Lanczos algorithm: Background

Consider the simpler case in which A is not time-dependent. The solution of (1.1) is given by the matrix function $\exp(A(t' - t))$ which can be numerically approximated in several different ways (see, e.g., [30–32]). One possible method is the (non-Hermitian) Lanczos algorithm. Computing the (k, ℓ) element of $\exp(A)$ is equivalent to computing the bilinear form $e_k^H \exp(A) e_\ell$, with e_k, e_ℓ vectors from the canonical Euclidean basis, and e_k^H the usual Hermitian transpose (here it coincides with the transpose since the vector is real). The non-Hermitian Lanczos algorithm (e.g., [33–36]) gives, when no breakdown occurs, the matrices

$$V_n = [v_0, \dots, v_{n-1}], \quad W_n = [w_0, \dots, w_{n-1}],$$

whose columns are biorthonormal bases respectively for the Krylov subspaces

$$\text{span}\{e_\ell, A e_\ell, \dots, A^{n-1} e_\ell\}, \quad \text{span}\{e_k, A^H e_k, \dots, (A^H)^{n-1} e_k\}.$$

Note that for A Hermitian and $k = \ell$ we can equivalently use the Hermitian Lanczos algorithm (getting $V_n = W_n$). The so-called (complex) *Jacobi matrix* J_n is the tridiagonal symmetric matrix with generally complex elements obtained by

$$J_n = W_n^H A V_n.$$

As described in [35], we can use the approximation

$$e_k^H \exp(A) e_\ell \approx e_1^H \exp(J_n) e_1, \quad (1.2)$$

which relies on the so-called *matching moment property*, i.e.,

$$e_k^H (A)^j e_\ell = e_1^H (J_n)^j e_1, \quad j = 0, 1, \dots, 2n - 1; \quad (1.3)$$

see, e.g., [35, 36] for the Hermitian case, and [37, 38] for the non-Hermitian one. The approximation (1.2) is a model reduction in two senses. First, the size of A is much larger than n – the size of J_n . Second, the structure of the matrix J_n is much simpler since it is tridiagonal. From a graph perspective, looking at A and J_n as adjacency matrices of, respectively, the graphs G and H_n , the possibly very complicated structure of G is reduced to the path (with self-loops) H_n . In this framework, Property (1.3) shows that the weighted number of walks in G of length j from the node k to the node ℓ is equal to the weighted number of closed walks of length j in H_n passing through the node 1, for $j = 0, 1, \dots, 2n - 1$; see, e.g., [39, 40].

Given a matrix A with size N , the Lanczos algorithm can be used as a method for its tridiagonalization (see, e.g., [41]). Assuming no breakdown, the N th iteration of the non-Hermitian Lanczos with input the matrix A and a couple of vectors v, w produces the tridiagonal matrix J_N , and the biorthogonal

square matrices V_N, W_N so that

$$A^j = V_N (J_N)^j W_N^H, \quad j = 0, 1, \dots, \quad (1.4)$$

giving the exact expression

$$\exp(A) = V_N \exp(J_N) W_N^H. \quad (1.5)$$

Theorem 2.3 which we prove in this work extends this result to time-ordered exponentials.

The Lanczos approximation (1.2) is connected with several further topics, such as (formal) orthogonal polynomials, Gauss quadrature, continued fractions, the moment problem, and many others. Information about these connections and references to the related rich and vast literature can be found, e.g., in the monographs [35, 36, 42] and the surveys [37, 38].

Inspired by approximation (1.2), the *-Lanczos algorithm produces a model reduction of a time-ordered exponential by providing a time-dependent tridiagonal matrix T_n satisfying properties analogous to the ones described above [17]. Differently from the classical case, the *-Lanczos algorithm works on vector distribution subspaces and it has to deal with a non-commutative product.

The time-dependent framework in which the proposed method works is much more complicated than the (time-independent) Krylov subspace approximation given by the (classical) Lanczos algorithm. In this paper, we will not deal with the behavior of the *-Lanczos algorithm considering approximations and finite-precision arithmetic problems.

1.3 Outline

The work is organized as follows: In Section 2, we build the *-Lanczos algorithm. The algorithm relies on a non-commutative *-product between generalized functions of two-time variables, which we describe in Section 2.1. Then, in Section 2.2, we state the main result, Theorem 2.2, which underpins the Lanczos-like procedure. The Theorem establishes that the first $2n$ *-moments of a certain tridiagonal matrix T_n match the corresponding *-moments of the original matrix A . Theorem 2.2 is proved with the tools developed in the subsequent Subsection 2.3. Section 3 is devoted to the convergence and breakdown properties of the algorithm, while examples of its use are given in Section 4. In Section 5 we outline a way to implement the Lanczos-like procedure numerically and we evaluate its computational cost. Section 6 concludes the paper.

2 The *-Lanczos Algorithm

2.1 The *-product and *-moments

In this section, we recall and complement with further properties and examples, the product introduced in [18, Section 1.2].

Let t and t' be two real variables. We consider the class $D(I)$ of all distributions which are linear superpositions of Heaviside theta functions and Dirac delta derivatives with smooth coefficients over I^2 . That is, a distribution d is in $D(I)$ if and only if it can be written as

$$d(t', t) = \tilde{d}(t', t)\Theta(t' - t) + \sum_{i=0}^N \tilde{d}_i(t', t)\delta^{(i)}(t' - t), \quad (2.1)$$

where $N \in \mathbb{N}$ is finite, $\Theta(\cdot)$ stands for the Heaviside theta function (with the convention $\Theta(0) = 1$) and $\delta^{(i)}(\cdot)$ is the i th derivative of the Dirac delta distribution $\delta = \delta^{(0)}$. Here and from now on, a tilde over a function (e.g., $\tilde{d}(t', t)$) indicates that it is an ordinary function *smooth* in both $t' \in I$ and $t \in I$. Note that we consider distributions as defined by Schwartz ([43, 44]). Hence a distribution $f \in D(I)$ should be interpreted as a linear functional applied to test functions.

We can endow the class $D(I)$ with a non-commutative algebraic structure upon defining a product between its elements. For $f_1, f_2 \in D(I)$ we define the convolution-like $*$ product between $f_1(t', t)$ and $f_2(t', t)$ as

$$(f_2 * f_1)(t', t) := \int_{-\infty}^{\infty} f_2(t', \tau)f_1(\tau, t) d\tau, \quad (2.2)$$

that has as identity element the Dirac delta distribution, $1_* := \delta(t' - t)$. When $f(t', t) = f(t' - t)$ has bounded supporting set, the $*$ -product $f * g$ (and $g * f$) is equivalent to the convolution product for distributions defined by Schwartz ([43, S 11] and [44, Chapter VI]). Since $\delta^{(i)}(t' - t)$ has bounded supporting set, given $f \in D(I)$, the $*$ -product $\delta^{(i)}(t' - t) * f$ and $f * \delta^{(i)}(t' - t)$ are well-defined and are both elements of $D(I)$; see [18] for further details. Moreover, it holds

$$\begin{aligned} \delta^{(i)}(t' - t) * \delta^{(j)}(t' - t) &= \delta^{(j)}(t' - t) * \delta^{(i)}(t' - t) = \delta^{(i+j)}(t' - t); \\ \Theta(t' - t) * \delta'(t' - t) &= \delta'(t' - t) * \Theta(t' - t) = \delta(t' - t). \end{aligned}$$

Consider the subclass $\text{Sm}_{\Theta}(I)$ of $D(I)$ comprising those distributions of the form

$$f(t', t) = \tilde{f}(t', t)\Theta(t' - t). \quad (2.3)$$

For $f_1, f_2 \in \text{Sm}_{\Theta}(I)$, the $*$ -product between f_1, f_2 simplifies to

$$(f_2 * f_1)(t', t) = \int_{-\infty}^{\infty} \tilde{f}_2(t', \tau)\tilde{f}_1(\tau, t)\Theta(t' - \tau)\Theta(\tau - t) d\tau,$$

$$= \Theta(t' - t) \int_t^{t'} \tilde{f}_2(t', \tau) \tilde{f}_1(\tau, t) d\tau,$$

which makes calculations involving such functions easier to carry out and shows that $\text{Sm}_\Theta(I)$ is closed under $*$ -multiplication. Together with the arguments above, this proves that $D(I)$ is closed under $*$ -multiplication. Hence, for $f \in D(I)$, we can define its k th $*$ -power f^{*k} as the k $*$ -products $f * f * \dots * f$, with the convention $f^{*0} = \delta(t' - t)$. First examples of $*$ -powers are

$$\Theta^{*k}(t' - t) = \frac{(t' - t)^{k-1}}{(k-1)!} \Theta(t' - t); \quad (2.4)$$

$$\left(\delta^{(j)}(t' - t)\right)^{*k} = \delta^{(kj)}(t' - t). \quad (2.5)$$

Note that, for members of $\text{Sm}_\Theta(I)$, the $*$ -product reduces exactly to the Volterra composition, a product between smooth functions of two-variables developed by Volterra and Pérès [45]. Volterra composition in the form above has seen little use since the 1950s because of perceived defects, such as the lack of identity element, which find remedies in the theory of distributions.

We illustrate the $*$ -product behavior with the following example.

Example 2.1 Let $f(t', t) = \tilde{f}(t', t)\Theta(t' - t) = 2 \sin(t')t\Theta(t' - t)$. Then

$$(f * \Theta(t' - t))(t', t) = \int_t^{t'} 2 \sin(t')\tau d\tau = \sin(t')(t'^2 - t^2)\Theta(t' - t),$$

is the integral of \tilde{f} with respect to t , while

$$(\Theta(t' - t) * f)(t', t) = \int_t^{t'} 2 \sin(\tau)t d\tau = 2t(\cos(t) - \cos(t'))\Theta(t' - t),$$

is the integral of \tilde{f} with respect to t' . On the other hand, the $*$ -products

$$\begin{aligned} (f * \delta'(t' - t))(t', t) &= \int_{-\infty}^{+\infty} 2 \sin(t')\tau \Theta(t' - \tau) \delta'(\tau - t) d\tau \\ &= -2 \sin(t')\Theta(t' - t) + 2 \sin(t')t' \delta(t' - t); \end{aligned}$$

$$\begin{aligned} (\delta'(t' - t) * f)(t', t) &= \int_{-\infty}^{+\infty} 2 \sin(\tau)t \Theta(\tau - t) \delta'(t' - \tau) d\tau \\ &= 2 \cos(t')t \Theta(t' - t) + 2 \sin(t)t \delta(t' - t); \end{aligned}$$

are derived by the formulas

$$(f * \delta'(t' - t))(t', t) = -\left(\frac{\partial}{\partial t} \tilde{f}(t', t)\right) \Theta(t' - t) + \tilde{f}(t', t') \delta(t' - t);$$

$$(\delta'(t' - t) * f)(t', t) = \left(\frac{\partial}{\partial t'} \tilde{f}(t', t)\right) \Theta(t' - t) + \tilde{f}(t, t) \delta(t' - t);$$

see [18, 44]. From here, one can verify that

$$(f * \delta'(t' - t)) * \Theta(t' - t) = f, \quad \Theta(t' - t) * (\delta'(t' - t) * f) = f.$$

8 *A Lanczos-like method for linear ODEs*

We will not discuss any further the $*$ -product by a Dirac delta derivative since it would bring us too far from the paper's goals. More details and examples can be found in [18].

The $*$ -product extends directly to distributions of $D(I)$ whose smooth coefficients depend on less than two variables. Indeed, consider a generalized function $f_3(t', t) = \tilde{f}_3(t')\delta^{(i)}(t' - t)$ with $i \geq -1$ and $\delta^{(-1)} = \Theta$. Then

$$(f_3 * f_1)(t', t) = \tilde{f}_3(t') \int_{-\infty}^{+\infty} \delta^{(i)}(t' - \tau) f_1(\tau, t) d\tau,$$

$$(f_1 * f_3)(t', t) = \int_{-\infty}^{+\infty} f_1(t', \tau) \tilde{f}_3(\tau) \delta^{(i)}(\tau - t) d\tau.$$

where $f_1(t', t)$ is defined as before. Hence the variable of $\tilde{f}_3(t')$ is treated as the left variable of a smooth function of two variables. This observation extends straightforwardly should \tilde{f}_3 be constant and, by linearity, to any distribution of $D(I)$.

The $*$ -product also naturally extends to matrices whose entries are distributions of $D(I)$. Consider two of such matrices $A_1(t', t)$ and $A_2(t', t) \in D(I)^{N \times N}$ then

$$(A_2 * A_1)(t', t) := \int_{-\infty}^{+\infty} A_2(t', \tau) A_1(\tau, t) d\tau,$$

where the sizes of A_1, A_2 are compatible for the usual matrix product (here and in the following, we omit the dependency on t' and t when it is clear from the context). As earlier, the $*$ -product is associative and distributive with respect to the addition, but it is non-commutative. The identity element with respect to this product is now $\text{ld}_* := \text{ld} 1_*$, with ld the identity matrix of appropriate size.

Given a square matrix $A(t', t)$ composed of elements from $D(I)$, we define the k -th *matrix $*$ -power* A^{*k} as the k $*$ -products $A * A * \dots * A$. In particular, by (2.4) we get the bound

$$\|A^{*k}(t', t)\|_* \leq \left(\sup_{\substack{\tau \geq \rho \\ \tau, \rho \in I}} \|A(\tau, \rho)\|_* \right)^k \frac{(t' - t)^{k-1}}{(k-1)!} \Theta(t' - t); \quad t', t \in I,$$

with $\|\cdot\|_*$ any induced matrix norm. As a consequence, the $*$ -resolvent of any matrix depending on at most two variables is well defined, as $R_*(A) := (\text{ld}_* - A)^{* -1} = \text{ld}_* + \sum_{k \geq 1} A^{*k}$ exists provided every entry of A is bounded for all $t', t \in I$ (see [26]). Then

$$U(t', t) = \Theta(t' - t) * R_*(A)(t', t) \tag{2.6}$$

is the time-ordered exponential of $A(t', t)$; see [26]. Note that time-ordered exponentials are usually presented with only one-time variable, corresponding to $U(t) = U(t, 0)$. Yet, in general $U(t', t) \neq U(t' - t, 0)$.

In the spirit of the Lanczos algorithm, given a time-dependent matrix $\mathbf{A}(t', t)$, we will construct a matrix $\mathbf{T}_n(t', t)$ of size $n \leq N$ with a simpler (tridiagonal) structure and so that, fixing the indexes k, ℓ , it holds

$$(\mathbf{A}^{*j}(t', t))_{k,\ell} = (\mathbf{T}_n^{*j}(t', t))_{1,1}, \quad \text{for } j = 0, \dots, 2n - 1, \quad t', t \in I; \quad (2.7)$$

compare it with (1.3). In particular, when $n = N$ Property (2.7) stands for every $j \geq 0$, giving

$$\mathbf{R}_*(\mathbf{A})_{k,\ell} = \mathbf{R}_*(\mathbf{T}_N)_{1,1}.$$

Hence the solution is given by the path-sum techniques exploiting the fact that the graph having \mathbf{T}_N as its adjacency matrix is a path that admits self-loops. More in general, given time-independent vectors \mathbf{v}, \mathbf{w} we call the j th **-moment* of $\mathbf{A}, \mathbf{v}, \mathbf{w}$ the scalar function $\mathbf{w}^H(\mathbf{A}^{*j}(t', t))\mathbf{v}$, for $j \geq 0$ (note that when the product is omitted, it stands for the usual matrix-vector product). Then Property (2.7) is an instance of the more general case

$$\mathbf{w}^H(\mathbf{A}^{*j}(t', t))\mathbf{v} = \mathbf{e}_1^H(\mathbf{T}_n^{*j}(t', t))\mathbf{e}_1, \quad \text{for } j = 0, \dots, 2n - 1, \quad t', t \in I.$$

2.2 Building up the *-Lanczos process

Given a doubly time-dependent matrix $\mathbf{A}(t', t) = \tilde{\mathbf{A}}(t')\Theta(t' - t)$ and $k + 1$ scalar generalized functions $\alpha_0(t', t), \alpha_1(t', t), \dots, \alpha_k(t', t) \in \mathbf{D}(I)$ which play the role of the *coefficients*, we define the *matrix *-polynomial* $p(\mathbf{A})(t', t)$ of degree k as

$$p(\mathbf{A})(t', t) := \sum_{j=0}^k (\mathbf{A}^{*j} * \alpha_j)(t', t);$$

moreover, we define the corresponding *dual* matrix *-polynomial as

$$p^D(\mathbf{A})(t', t) := \sum_{j=0}^k (\bar{\alpha}_j * (\mathbf{A}^{*j}))(t', t),$$

where, in general, \bar{d} is the conjugated value of $d \in \mathbf{D}(I)$ and it is defined by conjugating the functions \tilde{d} and \tilde{d}_i in (2.1). Let \mathbf{v} be a time independent vector, we can define the set of time-dependent vectors $p(\mathbf{A})\mathbf{v}$, with p a matrix *-polynomial. Such a set is a vector space with respect to the product $*$ and with scalars $\alpha_j(t', t)$ (the addition is the usual addition between vectors). Similarly, given a vector \mathbf{w}^H not depending on time, we can define the vector space given by the dual vectors $\mathbf{w}^H p^D(\mathbf{A})$. In particular, we can define the *-Krylov subspaces

$$\begin{aligned} \mathcal{K}_n(\mathbf{A}, \mathbf{v})(t', t) &:= \{ (p(\mathbf{A})\mathbf{v})(t', t) \mid p \text{ of degree at most } n - 1 \}, \\ \mathcal{K}_n^D(\mathbf{A}, \mathbf{w})(t', t) &:= \{ (\mathbf{w}^H p^D(\mathbf{A}))(t', t) \mid p \text{ of degree at most } n - 1 \}. \end{aligned}$$

The vectors $\mathbf{v}, \mathbf{A}\mathbf{v}, \dots, \mathbf{A}^{*(n-1)}\mathbf{v}$ and $\mathbf{w}^H, \mathbf{w}^H\mathbf{A}, \dots, \mathbf{w}^H\mathbf{A}^{*(n-1)}$ are bases respectively for $\mathcal{K}_n(\mathbf{A}, \mathbf{v})$ and $\mathcal{K}_n^D(\mathbf{A}, \mathbf{w})$. We aim to derive **-biorthonormal* bases $\mathbf{v}_0, \dots, \mathbf{v}_{n-1}$ and $\mathbf{w}_0^H, \dots, \mathbf{w}_{n-1}^H$ for the **-Krylov* subspaces, i.e., so that

$$\mathbf{w}_i^H * \mathbf{v}_j = \delta_{ij} \mathbf{1}_*, \quad (2.8)$$

with δ_{ij} the Kronecker delta.

Assume that $\mathbf{w}^H\mathbf{v} = 1$, we can use a non-Hermitian Lanczos like biorthogonalization process for the triplet $(\mathbf{w}, \mathbf{A}(t', t), \mathbf{v})$. We shall call this method the **-Lanczos process*. The first vectors of the biorthogonal bases are

$$\mathbf{v}_0 = \mathbf{v} \mathbf{1}_*, \quad \mathbf{w}_0^H = \mathbf{w}^H \mathbf{1}_*,$$

so that $\mathbf{w}_0^H * \mathbf{v}_0 = \mathbf{1}_*$. Consider now a vector $\widehat{\mathbf{v}}_1 \in \mathcal{K}_2(\mathbf{A}, \mathbf{v})$ given by

$$\widehat{\mathbf{v}}_1 = \mathbf{A} * \mathbf{v}_0 - \mathbf{v}_0 * \alpha_0 = \mathbf{A}\mathbf{v} - \mathbf{v}\alpha_0.$$

If the vector satisfies the **-biorthogonal* condition $\mathbf{w}_0^H * \widehat{\mathbf{v}}_1 = 0$, then

$$\alpha_0 = \mathbf{w}_0^H * \mathbf{A} * \mathbf{v}_0 = \mathbf{w}^H \mathbf{A} \mathbf{v}. \quad (2.9)$$

Similarly, we get the expression

$$\widehat{\mathbf{w}}_1^H = \mathbf{w}_0^H * \mathbf{A} - \alpha_0 * \mathbf{w}_0^H = \mathbf{w}^H \mathbf{A} - \alpha_0 \mathbf{w}^H,$$

with α_0 given by (2.9). Hence the **-biorthonormal* vectors are defined as

$$\mathbf{v}_1 = \widehat{\mathbf{v}}_1 * \beta_1^{*-1}, \quad \mathbf{w}_1 = \widehat{\mathbf{w}}_1,$$

with $\beta_1 = \widehat{\mathbf{w}}_1^H * \widehat{\mathbf{v}}_1$ and β_1^{*-1} its **-inverse*, i.e., $\beta_1^{*-1} * \beta_1 = \beta_1 * \beta_1^{*-1} = \mathbf{1}_*$. We give sufficient conditions for the existence of such **-inverses* below. Assume now that we have the **-biorthonormal* bases $\mathbf{v}_0, \dots, \mathbf{v}_{n-1}$ and $\mathbf{w}_0^H, \dots, \mathbf{w}_{n-1}^H$. Then we can build the vector

$$\widehat{\mathbf{v}}_n = \mathbf{A} * \mathbf{v}_{n-1} - \sum_{i=0}^{n-1} \mathbf{v}_i * \gamma_i,$$

where the γ_i are determined by the condition $\mathbf{w}_j^H * \widehat{\mathbf{v}}_n = \delta_{jn} \mathbf{1}_*$, for $j = 0, \dots, n-1$, giving

$$\gamma_j = \mathbf{w}_j^H * \mathbf{A} * \mathbf{v}_{n-1}, \quad j = 0, \dots, n-1.$$

In particular, since $\mathbf{w}_j^H * \mathbf{A} \in \mathcal{K}_{j+1}^D(\mathbf{A}, \mathbf{w})$ we get $\gamma_j = 0$ for $j = 0, \dots, n-3$. This leads to the following three-term recurrences for $n = 1, 2, \dots$ using the

convention $\mathbf{v}_{-1} = \mathbf{w}_{-1} = 0$,

$$\mathbf{w}_n^H = \mathbf{w}_{n-1}^H * \mathbf{A} - \alpha_{n-1} * \mathbf{w}_{n-1}^H - \beta_{n-1} * \mathbf{w}_{n-2}^H, \quad (2.10a)$$

$$\mathbf{v}_n * \beta_n = \mathbf{A} * \mathbf{v}_{n-1} - \mathbf{v}_{n-1} * \alpha_{n-1} - \mathbf{v}_{n-2}, \quad (2.10b)$$

with the coefficients given by

$$\alpha_{n-1} = \mathbf{w}_{n-1}^H * \mathbf{A} * \mathbf{v}_{n-1}, \quad \beta_n = \mathbf{w}_n^H * \mathbf{A} * \mathbf{v}_{n-1}. \quad (2.11)$$

Should β_n not be $*$ -invertible, we would get a *breakdown* in the algorithm, since it would be impossible to compute \mathbf{v}_n . We developed a range of general methods to determine the $*$ -inverse of functions of two-time variables which are gathered in [18]. These methods *constructively* show the existence of β_n^{*-1} almost everywhere on I under the following conditions:

- $\beta_n \in \text{Sm}_\Theta(I)$;
- $\beta_n \neq 0$ on I^2 .

Here the last condition means that β_n is not identically null over I^2 . The question of whether or not all $\alpha_n, \beta_n \in \text{Sm}_\Theta(I)$ was settled affirmatively in [17]. Let us define

$$\beta_n^{(1,0)}(t', t) := \frac{\partial}{\partial t} \beta_n(t', t), \quad t', t \in I.$$

In [17] we proved that if $\mathbf{A}(t', t) = \tilde{\mathbf{A}}(t')\Theta(t' - t)$, composed of elements from $\text{Sm}_\Theta(I)$, and if that $\beta_n^{(1,0)}(t, t) \neq 0$ for every $t \in I$, then the coefficients α_n, β_n are in $\text{Sm}_\Theta(I)$. The β_n are thus $*$ -invertible and, furthermore, their $*$ -inverses take on a particular form (see also Theorem 3.1 in Subsection 3.1). Moreover given $\rho \in I$, $\beta_n^{(1,0)}(\rho, \rho) \neq 0$ if and only if the (usual) non-Hermitian Lanczos algorithm does not breakdown when running on $\mathbf{A}(\rho), \mathbf{w}, \mathbf{v}$; see [17]. Since the issue of breakdowns of the $*$ -Lanczos algorithm is connected with the behavior of (usual) Lanczos techniques, we proceed as it is common when working with the non-Hermitian Lanczos algorithm. Thus, from now on, we assume all β_n to be $*$ -invertible and so that $\beta_n^{(1,0)}(t, t) \neq 0$ for every $t \in I$, while we come back to the issue of breakdowns in Section 3.2

The $*$ -orthogonalization process described above defines the $*$ -Lanczos algorithm (Table 1). The reason for this name is that the algorithm resembles the original Lanczos algorithm. Indeed, if all the inputs were *time-independent*, and if we substituted 1_* with 1 and the $*$ product with the usual matrix-vector or scalar-vector products, then Algorithm 1 would be mathematically equivalent to the non-Hermitian Lanczos algorithm.

Input: A complex time-dependent matrix $A = \widetilde{A}(t')\Theta(t' - t)$, and time-independent complex vectors \mathbf{v}, \mathbf{w} such that $\mathbf{w}^H \mathbf{v} = 1$.

Output: Vectors $\mathbf{v}_0, \dots, \mathbf{v}_{n-1}$ and vectors $\mathbf{w}_0, \dots, \mathbf{w}_{n-1}$ spanning respectively $\overline{\mathcal{K}_n(A, \mathbf{v})}$, $\mathcal{K}_n(A, \mathbf{w})$ and satisfying the *-biorthogonality conditions (2.8). The coefficients $\alpha_0, \dots, \alpha_{n-1}$ and β_1, \dots, β_n from the recurrences (2.10).

Initialize: $\mathbf{v}_{-1} = \mathbf{w}_{-1} = 0$, $\mathbf{v}_0 = \mathbf{v} 1_*$, $\mathbf{w}_0^H = \mathbf{w}^H 1_*$.

$\alpha_0 = \mathbf{w}^H A \mathbf{v}$,

$\mathbf{w}_1^H = \mathbf{w}^H A - \alpha_0 \mathbf{w}^H$,

$\widehat{\mathbf{v}}_1 = A \mathbf{v} - \alpha_0 \mathbf{v}$,

$\beta_1 = \mathbf{w}^H A^* \mathbf{v} - \alpha_0^*$,

If β_1 is not *-invertible, then stop, otherwise,

$\mathbf{v}_1 = \widehat{\mathbf{v}}_1 * \beta_1^{*-1}$,

For $n = 2, \dots$

$\alpha_{n-1} = \mathbf{w}_{n-1}^H * A * \mathbf{v}_{n-1}$,

$\mathbf{w}_n^H = \mathbf{w}_{n-1}^H * A - \alpha_{n-1} * \mathbf{w}_{n-1}^H - \beta_{n-1} * \mathbf{w}_{n-2}^H$,

$\widehat{\mathbf{v}}_n = A * \mathbf{v}_{n-1} - \mathbf{v}_{n-1} * \alpha_{n-1} - \mathbf{v}_{n-2}$,

$\beta_n = \mathbf{w}_n^H * A * \mathbf{v}_{n-1}$,

If β_n is not *-invertible, then stop, otherwise,

$\mathbf{v}_n = \widehat{\mathbf{v}}_n * \beta_n^{*-1}$,

end.

Table 1 *-Lanczos algorithm.

Let us define the tridiagonal matrix

$$\mathbf{T}_n := \begin{bmatrix} \alpha_0 & 1_* & & & \\ \beta_1 & \alpha_1 & \ddots & & \\ & \ddots & \ddots & & \\ & & & 1_* & \\ & & & \beta_{n-1} & \alpha_{n-1} \end{bmatrix}, \quad (2.12)$$

and the matrices $\mathbf{V}_n := [\mathbf{v}_0, \dots, \mathbf{v}_{n-1}]$ and $\mathbf{W}_n := [\mathbf{w}_0, \dots, \mathbf{w}_{n-1}]$. Then the three-term recurrences Eqs. (2.10) read, in matrix form,

$$A * \mathbf{V}_n = \mathbf{V}_n * \mathbf{T}_n + (\mathbf{v}_n * \beta_n) \mathbf{e}_n^H,$$

$$\mathbf{W}_n^H * A = \mathbf{T}_n * \mathbf{W}_n^H + \mathbf{e}_n \mathbf{w}_n^H.$$

Hence the tridiagonal matrix (2.12) can be expressed as

$$\mathbf{T}_n = \mathbf{W}_n^H * A * \mathbf{V}_n.$$

The following property of T_n is fundamental for the time-ordered exponential approximation. We will prove it in the following subsection.

Theorem 2.2 (Matching Moment Property) *Let A , \mathbf{w} , \mathbf{v} and T_n be as described above, then*

$$\mathbf{w}^H (\mathsf{A}^{*j}) \mathbf{v} = \mathbf{e}_1^H (\mathsf{T}_n^{*j}) \mathbf{e}_1, \quad \text{for } j = 0, \dots, 2n - 1. \quad (2.13)$$

Consider the time-ordered exponential U_n given by the differential equation

$$\mathsf{T}_n(t', t) \mathsf{U}_n(t', t) = \frac{d}{dt'} \mathsf{U}_n(t', t), \quad \mathsf{U}_n(t, t) = \text{Id}. \quad (2.14)$$

Theorem 2.2 and Eq. (2.6) justify the use of the approximation

$$\mathbf{w}^H \mathsf{U}(t', t) \mathbf{v} \approx \mathbf{e}_1^H \mathsf{U}_n(t', t) \mathbf{e}_1 = \Theta(t' - t) \int_t^{t'} \mathsf{R}_*(\mathsf{T}_n)_{1,1}(\tau, t) d\tau; \quad (2.15)$$

see [17]. The system (2.14) can be seen as a reduced order model of the initial differential Eq. (1.1) from two points of view. First, n may be much smaller than the size of A ; in this sense, in Section 3, we will discuss the convergence behavior of the approximation using Theorem 2.2. Secondly, looking at A and T_n as adjacency matrices, A may correspond to a graph with a complex structure, while T_n corresponds to a very simple graph composed of one path with possible self-loops. Then the path-sum method gives

$$\mathsf{R}_*(\mathsf{T}_n)_{1,1}(t', t) = \left(1_* - \alpha_0 - (1_* - \alpha_1 - (1_* - \dots)^{* - 1} * \beta_2)^{* - 1} * \beta_1 \right)^{* - 1}, \quad (2.16)$$

see [26, 46]. This expression is analogous to the one for the first diagonal entry of the inverse of an ordinary tridiagonal matrix [47] (see also [35, 36] for Jacobi matrices), except here all operations are taken with respect to the $*$ -product.

For $n = N$, we get

$$\mathsf{V}_N * \mathsf{W}_N^H = \mathsf{W}_N^H * \mathsf{V}_N = \text{Id } 1_*. \quad (2.17)$$

As a consequence

$$\mathbf{w}^H (\mathsf{A}^{*j}) \mathbf{v} = \mathbf{e}_1^H (\mathsf{T}_N^{*j}) \mathbf{e}_1, \quad \text{for } j = 0, 1, \dots, \quad (2.18)$$

and therefore the approximation (2.15) is actually exact. More generally, for $n = N$, the $*$ -Lanczos algorithm produces the full tridiagonalization of A and of its time-ordered exponential, in a manner analogous to (1.4) and (1.5); see [17].

Theorem 2.3 *Let $\mathbf{A}, \mathbf{V}_N, \mathbf{W}_N$ and \mathbf{T}_N be as described above, then*

$$\mathbf{A}^{*j} = \mathbf{V}_N * \mathbf{T}_N^{*j} * \mathbf{W}_N^H, \quad j = 0, 1, \dots,$$

and thus

$$\mathbf{R}_*(\mathbf{A}) = \mathbf{V}_N * \mathbf{R}_*(\mathbf{T}_N) * \mathbf{W}_N^H.$$

The theorem follows by using (2.17). Here, any entry of $\mathbf{R}_*(\mathbf{T}_N)$ is calculable using a path-sum continued of depth at most N .

Remark 2.4 The Lanczos-like method presented here for the time-ordered exponential is immediately valid for the ordinary matrix exponential function, since the latter is obtained from the former in the situation where \mathbf{A} commutes with itself at all times,

$$\mathcal{T}e^{\int A(\tau) d\tau} = e^{\int_t^{t'} A(\tau) d\tau}.$$

This situation includes the case where \mathbf{A} is time-independent, in which case setting $t = 0$ and $t' = 1$ above yields the matrix exponential of \mathbf{A} . However, the $*$ -Lanczos algorithm cannot be considered a generalization of the Lanczos algorithm since its outputs on constant matrices are made of distributions and time dependent functions.

2.3 Matching $*$ -moments through $*$ -biorthonormal polynomials

In order to prove Theorem 2.2, we will exploit the connection between the $*$ -Lanczos algorithm and families of $*$ -biorthonormal polynomials. Let us define the set of $*$ -polynomials

$$\mathcal{P}_* := \left\{ p(\lambda) = \sum_{j=0}^k \lambda^{*j} * \gamma_j(t', t) \right\},$$

with $\gamma_j(t', t) \in \mathbf{D}(I)$. Consider a $*$ -sesquilinear form $[\cdot, \cdot] : \mathcal{P}_* \times \mathcal{P}_* \rightarrow \mathbf{D}(I)$, i.e., so that given $p_1, p_2, q_1, q_2 \in \mathcal{P}_*$ and $\alpha, \beta \in \mathbf{D}(I)$, it satisfies

$$\begin{aligned} [q_1 * \alpha, p_1 * \beta] &= \bar{\alpha} * [q_1, p_1] * \beta, \\ [q_1 + q_2, p_1 + p_2] &= [q_1, p_1] + [q_2, p_1] + [q_1, p_2] + [q_2, p_2]. \end{aligned}$$

From now on we assume that every considered $*$ -sesquilinear form $[\cdot, \cdot]$ also satisfies

$$[\lambda * q, p] = [q, \lambda * p]. \quad (2.19)$$

The $*$ -sesquilinear form $[\cdot, \cdot]$ is determined by its $*$ -moments defined as

$$m_j(t, t') := [\lambda^{*j}, 1] = [1, \lambda^{*j}], \quad j = 0, 1, \dots$$

We aim to build sequences of $*$ -polynomials p_0, p_1, \dots and q_0, q_1, \dots so that they are $*$ -biorthonormal with respect to $[\cdot, \cdot]$, i.e.,

$$[q_i, p_j] = \delta_{ij} 1_*, \quad (2.20)$$

where the subindex j in p_j and q_j corresponds to the degree of the $*$ -polynomial. Here and in the following we assume $m_0 = 1_*$, getting $p_0 = q_0 = 1_*$. Consider the $*$ -polynomial

$$q_1(\lambda) = \lambda * q_0(\lambda) - q_0(\lambda) * \bar{\alpha}_0.$$

The orthogonality conditions (2.20) give $\alpha_0 = [\lambda * q_0, p_0]$. Similarly, we get the $*$ -polynomial

$$p_1(\lambda) * \beta_1 = \lambda * p_0(\lambda) - p_0(\lambda) * \alpha_0,$$

with $\alpha_0 = [q_0, \lambda * p_0]$, $\beta_1 = [q_1, \lambda * p_0]$. Repeating the $*$ -orthogonalization process, we obtain the three-term recurrences for $n = 1, 2, \dots$

$$q_n(\lambda) = \lambda * q_{n-1}(\lambda) - q_{n-1}(\lambda) * \bar{\alpha}_{n-1} - q_{n-2}(\lambda) * \bar{\beta}_{n-1} \quad (2.21a)$$

$$p_n(\lambda) * \beta_n = \lambda * p_{n-1}(\lambda) - p_{n-1}(\lambda) * \alpha_{n-1} - p_{n-2}(\lambda), \quad (2.21b)$$

with $p_{-1} = q_{-1} = 0$ and

$$\alpha_{n-1} = [q_{n-1}, \lambda * p_{n-1}], \quad \beta_n = [q_n, \lambda * p_{n-1}]. \quad (2.22)$$

Note that deriving the recurrences needs property (2.19). The $*$ -biorthonormal polynomials p_0, \dots, p_n and q_0, \dots, q_n exist under the assumption that β_1, \dots, β_n are $*$ -invertible.

Let \mathbf{A} be a time-dependent matrix, and \mathbf{w}, \mathbf{v} time-independent vectors such that $\mathbf{w}^H \mathbf{v} = 1$. Consider the $*$ -sesquilinear form $[\cdot, \cdot]$ defined by

$$[q, p] = \mathbf{w}^H q^D(\mathbf{A}) * p(\mathbf{A}) \mathbf{v}.$$

Assume that there exist $*$ -polynomials p_0, \dots, p_n and q_0, \dots, q_n $*$ -biorthonormal with respect to $[\cdot, \cdot]$. Defining the vectors

$$\mathbf{v}_j = p_j(\mathbf{A}) \mathbf{v}, \quad \mathbf{w}_j^H = \mathbf{w}^H q_j^D(\mathbf{A}),$$

and using the recurrences (2.21) gives the $*$ -Lanczos recurrences (2.10). Moreover, the coefficients in (2.22) are the $*$ -Lanczos coefficients in (2.11).

Let \mathbf{T}_n be a tridiagonal matrix as in (2.12) composed of the coefficients (2.22) associated with the $*$ -sesquilinear form $[\cdot, \cdot]$. Then we can define the $*$ -sesquilinear form

$$[q, p]_n = \mathbf{e}_1^H q^D(\mathbf{T}_n) * p(\mathbf{T}_n) \mathbf{e}_1.$$

The following lemmas show that

$$m_j = [\lambda^{*j}, 1_*] = [\lambda^{*j}, 1_*]_n, \quad j = 0, \dots, 2n - 1,$$

proving Theorem 2.2.

Lemma 2.5 Let p_0, \dots, p_n and q_0, \dots, q_n be $*$ -biorthonormal polynomials with respect to the $*$ -sesquilinear form $[\cdot, \cdot]$. Assume that the coefficients β_1, \dots, β_n in the related recurrences (2.21) are $*$ -invertible. Then the $*$ -polynomials are also $*$ -biorthonormal with respect to the form $[\cdot, \cdot]_n$ defined above.

Proof Consider the vectors $\mathbf{y}_j^H = \mathbf{e}_1^H \mathbb{T}_n^{*j}$ and $\mathbf{x}_j = \mathbb{T}_n^{*j} \mathbf{e}_1$. Since the matrix \mathbb{T}_n is tridiagonal, for $j = 1, \dots, n-1$, we have

$$\begin{aligned} \mathbf{e}_i^H \mathbf{x}_j &= 0, \text{ for } i \geq j+2, & \text{and} & & \mathbf{e}_{j+1}^H \mathbf{x}_j &= \beta_j * \dots * \beta_1, \\ \mathbf{y}_j^H \mathbf{e}_i &= 0, \text{ for } i \geq j+2, & \text{and} & & \mathbf{y}_j^H \mathbf{e}_{j+1} &= 1_*. \end{aligned}$$

By assumption, the product $\beta_j * \dots * \beta_1$ is $*$ -invertible. Therefore there exist $*$ -polynomials $\widehat{p}_0, \dots, \widehat{p}_{n-1}$ and $\widehat{q}_0, \dots, \widehat{q}_{n-1}$ so that, for $i = 0, \dots, n-1$, we get

$$1_* \mathbf{e}_{i+1}^H = \mathbf{e}_1^H \widehat{q}_i^D(\mathbb{T}_n), \quad 1_* \mathbf{e}_{i+1} = \widehat{p}_i(\mathbb{T}_n) \mathbf{e}_1.$$

Such $*$ -polynomials are $*$ -biorthonormal with respect to $[\cdot, \cdot]_n$ since they satisfy

$$[\widehat{q}_i, \widehat{p}_j]_n = 1_* \mathbf{e}_{i+1}^H * 1_* \mathbf{e}_{j+1} = \delta_{ij} 1_*.$$

Moreover, for $i = 0, \dots, n-1$, the corresponding recurrence coefficients (2.22) are the same as the ones of the $*$ -polynomials p_0, \dots, p_{n-1} and q_0, \dots, q_{n-1} . Indeed,

$$\begin{aligned} \widehat{\alpha}_{i-1} &= [\widehat{q}_{i-1}, \lambda * \widehat{p}_{i-1}]_n = \mathbf{e}_{i-1}^H \mathbb{T}_n \mathbf{e}_{i-1} = \alpha_{i-1}, \\ \widehat{\beta}_i &= [\widehat{q}_i, \lambda * \widehat{p}_{i-1}]_n = \mathbf{e}_i^H \mathbb{T}_n \mathbf{e}_{i-1} = \beta_i. \end{aligned}$$

Since $\widehat{p}_0 = p_0 = \widehat{q}_0 = q_0 = 1_*$, we get $\widehat{p}_i = p_i$ and $\widehat{q}_i = q_i$ for $i = 0, \dots, n-1$. \square

Lemma 2.6 Let p_0, \dots, p_{n-1} and q_0, \dots, q_{n-1} be $*$ -biorthonormal polynomials with respect to a $*$ -sesquilinear form $[\cdot, \cdot]_A$ and to a $*$ -sesquilinear form $[\cdot, \cdot]_B$. If $[1_*, 1_*]_A = [1_*, 1_*]_B = 1_*$, then $[\lambda^{*j}, 1_*]_A = [\lambda^{*j}, 1_*]_B$ for $j = 0, \dots, 2n-1$.

Proof We prove it by induction. Let $m_j = [\lambda^{*j}, 1_*]_A$ and $\widehat{m}_j = [\lambda^{*j}, 1_*]_B$ for $j = 0, 1, \dots, 2n-1$. By the expression for the coefficients in (2.22) we get

$$[q_0, \lambda * p_0]_A = \alpha_0 = [q_0, \lambda * p_0]_B.$$

Hence $m_1 = \alpha_0 = \widehat{m}_1$. Assuming $m_j = \widehat{m}_j$ for $j = 0, \dots, 2k-3$ we will prove that $m_{2k-2} = \widehat{m}_{2k-2}$ and $m_{2k-1} = \widehat{m}_{2k-1}$, for $k = 2, \dots, n$. The coefficient expressions in (2.22) gives

$$[q_{k-1}, \lambda * p_{k-2}]_A = \beta_{k-1} = [q_{k-1}, \lambda * p_{k-2}]_B,$$

which can be rewritten as

$$\sum_{i=0}^{k-1} \sum_{j=0}^{k-2} \bar{a}_i * m_{i+j+1} * b_j = \sum_{i=0}^{k-1} \sum_{j=0}^{k-2} \bar{a}_i * \widehat{m}_{i+j+1} * b_j,$$

with \bar{a}_i, b_j the coefficients respectively of q_{k-1} and p_{k-2} . The inductive assumption implies

$$\bar{a}_{k-1} * m_{2k-2} * b_{k-2} = \bar{a}_{k-1} * \widehat{m}_{2k-2} * b_{k-2}.$$

The leading coefficients of the $*$ -polynomials q_{2k-2} and p_{2k-2} are respectively $a_{k-1} = 1_*$ and $b_{k-2} = (\beta_{k-2} * \dots * \beta_1)^{*^{-1}}$. Hence $m_{2k-2} = \widehat{m}_{2k-2}$. Repeating the same argument with the coefficient α_{k-1} (2.22) concludes the proof. \square

3 Convergence, breakdown, and related properties

3.1 The convergence behavior of intermediate approximations

Assuming no breakdown, the *-Lanczos algorithm in conjunction with the path-sum method converges to the solution $\mathbf{w}^H \mathbf{U}(t', t) \mathbf{v}$ in N iterations, with N the size of \mathbf{A} ; see Eq. (2.18). Most importantly, intermediate *-Lanczos iterations provide a sequence of approximations $\int_t^{t'} \mathbf{R}_*(\mathbb{T}_n)_{1,1}(\tau, t) d\tau$, $n = 1, \dots, N$, whose convergence behavior we analyze hereafter.

As we have already discussed before, under the assumption that all entries of \mathbf{A} are smooth over I and that all the $\beta_j^{(1,0)}(t, t) \neq 0$, for every $t \in I$, all the α_j and β_j distributions are elements of $\text{Sm}_\Theta(I)$. The proof of this statement is very long and technical and serves only to establish the theoretical *feasibility* of tridagonalization for systems of coupled linear differential equations with variable coefficients using smooth functions. It was therefore presented in a separate work. We refer the reader to [17] for a full exposition and proof, while here we only state some of the main results:

Theorem 3.1 (P.-L. G. and S. P. [17]) *Let $\mathbf{A}(t', t) = \tilde{\mathbf{A}}(t')\Theta(t' - t)$ be an $N \times N$ matrix composed of elements from $\text{Sm}_\Theta(I)$. Let α_{n-1} and β_n be the coefficients generated by Algorithm 1 running on \mathbf{A} and the time-independent vectors \mathbf{w}, \mathbf{v} ($\mathbf{w}^H \mathbf{v} = 1$). For any $1 \leq n \leq N$, assuming that $\beta_j^{(1,0)}(t, t) \neq 0$ for every $t \in I$, $j = 1, \dots, n-1$, we get $\beta_n(t', t), \alpha_{n-1}(t', t) \in \text{Sm}_\Theta(I)$. In addition, all required *-inverses β_j^{*-1} exist and are of the form*

$$\beta_j^{*-1} = b_j^L(t', t) * \delta^{(3)}(t' - t),$$

with $b_j^L \in \text{Sm}_\Theta(I)$.

All b_j^L have explicit expansions in terms of β_j which are given in [17] but not reproduced here owing to length concerns. When $\beta_n \neq 0$, but $\beta_n^{(1,0)}(\rho, \rho) = 0$ for some $\rho \in I$, the results of Theorem 3.1 hold if we restrict the initial interval I to a subinterval $J \subset I$ so that $\rho \notin J$.

Thanks to these regularity results, we can establish the following bound for the approximation error:

Proposition 3.2 *Let us consider the setting and assumptions of Theorem 3.1. Moreover, let \mathbf{U} designate the time-ordered exponential of \mathbf{A} and let \mathbb{T}_n be the tridiagonal matrix (2.12) such that*

$$\mathbf{w}^H \mathbf{A}^{*j} \mathbf{v} = (\mathbb{T}_n^{*j})_{1,1}, \quad \text{for } j = 0, \dots, 2n-1.$$

Then, for $t' \geq t$, with $t', t \in I$,

$$\left| \mathbf{w}^H \mathbf{U}(t', t) \mathbf{v} - \int_t^{t'} \mathbf{R}_*(\mathbb{T}_n)_{1,1}(\tau, t) d\tau \right| \leq \frac{C^{2n} + D_n^{2n}}{(2n)!} (t' - t)^{2n} e^{(C+D_n)(t'-t)}.$$

Here

$$C := \sup_{t' \in I} \|\mathbf{A}(t')\|_\infty, \quad D_n := 3 \sup_{t', t \in I^2} \max_{0 \leq j \leq n-1} \{|\alpha_j(t', t)|, |\beta_j(t', t)|\}$$

are both finite, with $\|\cdot\|_\infty$ the matrix norm induced by the uniform norm..

Proof Assume $t' \geq t$. Observe that

$$\mathbf{w}^H \mathbf{U}(t', t) \mathbf{v} - \int_t^{t'} \mathbf{R}_*(\mathbb{T}_n)(\tau, t)_{11} \, d\tau = \int_t^{t'} \sum_{j=2n}^{\infty} \mathbf{w}^H \mathbf{A}^{*j}(\tau, t) \mathbf{v} - (\mathbb{T}_n^{*j})_{1,1}(\tau, t) \, d\tau,$$

so that

$$\left| \mathbf{w}^H \mathbf{U}(t', t) \mathbf{v} - \int_t^{t'} \mathbf{R}_*(\mathbb{T}_n)(\tau, t)_{11} \, d\tau \right| \leq \int_t^{t'} \sum_{j=2n}^{\infty} \left| \mathbf{w}^H \mathbf{A}^{*j}(\tau, t) \mathbf{v} \right| + \left| (\mathbb{T}_n^{*j})_{1,1}(\tau, t) \right| \, d\tau.$$

Now $\sup_{t' \in I} |\mathbf{w}^H \mathbf{A}(t') \mathbf{v}| \leq C$ and

$$\left| \int_t^{t'} \mathbf{w}^H \mathbf{A}^{*j}(\tau, t) \mathbf{v} \, d\tau \right| \leq \Theta(t' - t) * (C\Theta(t' - t))^{*j} = C^j \frac{(t' - t)^j}{j!}.$$

We proceed similarly for the terms involving \mathbb{T}_n . Theorem 3.1 implies the existence of $\widehat{D}_n := \sup_{t', t \in I^2} \max_{0 \leq j \leq n-1} \{|\alpha_j(t', t)|, |\beta_j(t', t)|\} < +\infty$. The matrix element $(\mathbb{T}_n^{*j})_{1,1}$ is given by the sum of $*$ -products of coefficients α_i, β_i and 1_* . Replacing all the factors in those $*$ -products with $\widehat{D}_n \Theta(t' - t)$ gives an upper bound for $(\mathbb{T}_n^{*j})_{1,1}$. Hence we get

$$\left| (\mathbb{T}_n^{*j})_{1,1} \right| \leq \left(\left(\widehat{D}_n \mathbf{P}_n \Theta(t' - t) \right)^{*j} \right)_{1,1} \leq \widehat{D}_n^j \|\mathbf{P}_n\|_\infty^j \Theta(t' - t)^{*j},$$

where \mathbf{P}_n is the $n \times n$ tridiagonal matrix whose nonzero entries are equal to 1. Note that $\|\mathbf{P}_n\|_\infty = 3$. Hence the error can be bounded by

$$\begin{aligned} \sum_{j=2n}^{\infty} (C^j + D_n^j) \frac{(t' - t)^j}{j!} &\leq \frac{(C^{2n} + D_n^{2n})}{2n!} (t' - t)^{2n} \sum_{j=0}^{\infty} \frac{2n!}{(2n + j)!} (C^j + D_n^j) (t' - t)^j, \\ &\leq \frac{(C^{2n} + D_n^{2n})}{2n!} (t' - t)^{2n} \sum_{j=0}^{\infty} \frac{(C + D_n)^j (t' - t)^j}{j!}, \\ &\leq \frac{(C^{2n} + D_n^{2n})}{2n!} (t' - t)^{2n} e^{(C+D_n)(t'-t)}, \end{aligned}$$

concluding the proof. \square

Analogously to the classical non-Hermitian Lanczos algorithm, we need to further assume D_n to be not too large for $n \geq 1$ in order to get a meaningful bound. Such an assumption can be verified a-posteriori. The bound of Proposition 3.2 demonstrates that under reasonable assumptions, the approximation error has a super-linear decay. Assuming no breakdown, we also recall that the algorithm does necessarily converge in at most N steps, independently of the error bound. The computational cost of the algorithm is discussed separately in Section 5.

3.2 Breakdown

In the classical non-Hermitian Lanczos algorithm, a breakdown appears either when an invariant Krylov subspace is produced (*lucky breakdown*) or when the last vectors of the biorthogonal bases $\mathbf{v}_n, \mathbf{w}_n$ are nonzero, but $\mathbf{w}_n^H \mathbf{v}_n = 0$ (*serious breakdown*); for further details refer, e.g., to [33, 34, 41, 48–50]. Analogously, in the *-Lanczos algorithm 1, a *lucky breakdown* arises when either $\mathbf{w}_n \equiv 0$ or $\widehat{\mathbf{v}}_n \equiv 0$. In such a case, the algorithm has converged to the solution, as the following proposition shows.

Proposition 3.3 *Assume that the *-Lanczos algorithm in Table 1 does not breakdown until the n th step when a lucky breakdown arises, i.e., $\widehat{\mathbf{v}}_n \equiv 0$ (or $\mathbf{w}_n \equiv 0$). Then*

$$\begin{aligned} \mathbf{w}^H(\mathbf{A}^{*j})\mathbf{v} &= \mathbf{e}_1^H(\mathbb{T}_n^{*j})\mathbf{e}_1, \quad \text{for } j \geq 0, \\ \mathbf{w}^H\mathbf{U}(t', t)\mathbf{v} &= \int_0^t \mathbf{R}_*(\mathbb{T}_n)_{1,1}(\tau, t) \, d\tau. \end{aligned}$$

Proof We prove it for $\widehat{\mathbf{v}}_n \equiv 0$. The case $\mathbf{w}_n \equiv 0$ follows similarly. By the results in Subsection 2.3, there exists a *-polynomial $\widehat{p}_n(\lambda) = \sum_{j=0}^n \lambda^{*j} * \gamma_j$, so that $\widehat{p}_n(\mathbf{A})\mathbf{v} = \widehat{\mathbf{v}}_n \equiv 0$. Therefore

$$\mathbf{A}^{*n}\mathbf{v} = - \sum_{j=0}^{n-1} \mathbf{A}^{*j}\mathbf{v} * \gamma_j.$$

Hence, in general, for every $k \geq n$ there exist a *-polynomial r_{n-1} of degree $n-1$ so that $\mathbf{A}^{*k}\mathbf{v} = r_{n-1}(\mathbf{A})\mathbf{v}$.

By Theorem 2.2 and using the notation of Subsection 2.3, we get

$$\mathbf{e}_1^H q_j^D(\mathbb{T}_n) * \widehat{p}_n(\mathbb{T}_n) \mathbf{e}_1 = \mathbf{w}^H q_j^D(\mathbf{A}) * \widehat{p}_n(\mathbf{A}) \mathbf{v} = 0, \quad j = 0, \dots, n-1.$$

As shown in the proof of Lemma 2.5, $\mathbf{e}_1^H q_j^D(\mathbb{T}_n) = 1 * \mathbf{e}_{j+1}^H$, for $j = 0, \dots, n-1$. Therefore we get $\widehat{p}_n(\mathbb{T}_n) \mathbf{e}_1 = 0$. As a consequence, for every $k \geq n$ we get

$$(\mathbb{T}_n)^{*n} \mathbf{e}_1 = - \sum_{j=0}^{n-1} (\mathbb{T}_n)^{*j} \mathbf{e}_1 * \gamma_j,$$

and thus $(\mathbb{T}_n)^{*k} \mathbf{e}_1 = r_{n-1}(\mathbb{T}_n) \mathbf{e}_1$. Since r_{n-1} has degree $n-1$, Theorem 2.2 implies

$$\mathbf{w}^H \mathbf{A}^{*k} \mathbf{v} = \mathbf{w}^H r_{n-1}(\mathbf{A}) \mathbf{v} = \mathbf{e}_1^H r_{n-1}(\mathbb{T}_n) \mathbf{e}_1 = \mathbf{e}_1^H (\mathbb{T}_n)^{*k} \mathbf{e}_1,$$

concluding the proof. \square

The *-Lanczos algorithm construction and its polynomial interpretation in Subsection 2.3 suggest that it may be possible to deal with the serious breakdown issue by a look-ahead strategy analogous to the one for the non-Hermitian Lanczos algorithm; see, e.g., [33, 34, 41, 48–53]. Nevertheless, we need to discuss the particular case of serious breakdowns arising when $\mathbf{w} = \mathbf{e}_i$ and $\mathbf{v} = \mathbf{e}_j$. If $i \neq j$, then $\mathbf{w}^H \mathbf{v} = 0$, which does not satisfy the *-Lanczos assumption. Moreover, if $i = j$ and \mathbf{A} is a sparse non-Hermitian matrix, then it may be possible that $\mathbf{A}_{ii} \equiv 0$ and $\mathbf{A}^{*2}_{ii} \equiv 0$. As a consequence, we get

$\beta_1 \equiv 0$. We can try to fix these problems rewriting the approximation of the time-ordered exponential \mathbf{U} as

$$e_i^H \mathbf{U} e_j = (e + e_i)^H \mathbf{U} e_j - e^H \mathbf{U} e_j,$$

with $\mathbf{e} = (1, \dots, 1)^H$. Then one can approximate $(e + e_i)^H \mathbf{U} e_j$ and $e^H \mathbf{U} e_j$ separately, which are less likely going to have a breakdown, thanks to the fact that \mathbf{e} is a full vector; see, e.g., [35, Section 7.3].

4 Examples

In this section, we use the *-Lanczos algorithm 1 on examples in ascending order of difficulty. All the computations have been performed using MATHEMATICA 11.

Example 4.1 (Ordinary matrix exponential) Let us first consider a constant matrix

$$\mathbf{A} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & -1 \end{pmatrix}.$$

Because \mathbf{A} commutes with itself at all times, its time-ordered exponential coincides with its ordinary exponential, $\mathcal{T}e^{\int \mathbf{A}(\tau) d\tau} \equiv e^{\mathbf{A}(t')}$ (we set $t = 0$). Note that the matrix chosen here is symmetric only to lead to concise expressions suitable for presentation in an article, e.g.,

$$(e^{\mathbf{A}t'})_{11} = -\frac{1}{2} \sinh(2t') + \frac{1}{2} \cosh(2t') + \frac{1}{2} \cosh(\sqrt{2}t'), \quad (4.1)$$

and that such symmetries are not a requirement of the *-Lanczos approach.

Now let us find the result of Eq. (4.1) with Algorithm 1. We define $\mathbf{w} := \mathbf{v}^H := (1, 0, 0)$, $\mathbf{w}_0 = \mathbf{w}1_*$, $\mathbf{v}_0 = \mathbf{v}1_*$, from which it follows that $\alpha_0(t', t) = -1 \times \Theta(t' - t)$ and $\mathbf{w}_1 = \widehat{\mathbf{v}}_1^H = (0, 1, 1)\Theta(t' - t)$. Furthermore, since \mathbf{A} is a constant matrix times $\Theta(t' - t)$, we have $\mathbf{A}^{*n} = \tilde{\mathbf{A}}^n \times \Theta(t' - t)^{*n} = \tilde{\mathbf{A}} \times (t' - t)^{n-1} / (n-1)! \times \Theta(t' - t)$ and similarly $\alpha_0^{*2}(t', t) = \tilde{\alpha}_0^2 \times (t' - t)\Theta(t' - t)$. Thus

$$\beta_1 = \mathbf{w}^H \mathbf{A}^2 \mathbf{v} \times (t' - t)\Theta(t' - t) - \tilde{\alpha}_0^2(t' - t)\Theta(t' - t) = 2(t' - t)\Theta(t' - t).$$

The *-inverse follows as $\beta^{*-1} = \frac{1}{2} \delta''$ [18], from which we get

$$\mathbf{v}_1 = \widehat{\mathbf{v}}_1 * \beta_1^{*-1} = (0, 1, 1)^H \frac{1}{2} \delta'(t' - t),$$

Now it follows that

$$\alpha_1(t', t) = \mathbf{w}_1 * \mathbf{A} * \mathbf{v}_1 = \frac{1}{2} \Theta(t - t'),$$

$$\mathbf{w}_2(t', t) = \mathbf{w}_1 * \mathbf{A} - \alpha_1 * \mathbf{w}_1 - \beta_1 * \mathbf{w}_0 = (0, 1, -1) \frac{1}{2} (t' - t)\Theta(t' - t),$$

$$\widehat{\mathbf{v}}_2(t', t) = \mathbf{A} * \mathbf{v}_1 - \mathbf{v}_1 * \alpha_1 - \mathbf{v}_0 = (0, 1, -1)^H \frac{1}{4} \delta(t' - t),$$

$$\beta_2 = \mathbf{w}_2 * \mathbf{A} * \mathbf{v}_1 = \frac{1}{4} (t' - t)\Theta(t' - t).$$

Then $\beta_2^{*-1} = 4\delta''$ and so

$$\mathbf{v}_2 = \widehat{\mathbf{v}}_2 * \beta_2^{*-1} = (0, 1, -1)^H \delta''(t' - t) \quad \alpha_2 = \mathbf{w}_2 * \mathbf{A} * \mathbf{v}_2 = -\frac{3}{2}\Theta(t' - t).$$

At this point we have determined the *-Lanczos matrices \mathbf{T} , \mathbf{V} and \mathbf{W} entirely

$$\mathbf{T} = \begin{pmatrix} -\Theta & \delta & 0 \\ 2\Theta^{*2} & \frac{1}{2}\Theta & \delta \\ 0 & \frac{1}{4}\Theta^{*2} & -\frac{3}{2}\Theta \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} \delta & 0 & 0 \\ 0 & \frac{1}{2}\delta' & \delta'' \\ 0 & \frac{1}{2}\delta' & -\delta'' \end{pmatrix}, \quad \mathbf{W}^H = \begin{pmatrix} \delta & 0 & 0 \\ 0 & \Theta & \Theta \\ 0 & \frac{1}{2}\Theta^{*2} & -\frac{1}{2}\Theta^{*2} \end{pmatrix}.$$

In all of these expressions, Θ is a short-hand notation for $\Theta(t' - t)$ and δ , δ' and δ'' are to be evaluated in $t' - t$. It is now straightforward to verify the matching moment property $(\mathbf{T}^{*j})_{11} = (\mathbf{A}^{*j})_{11}$ for all $j \in \mathbb{N}$. We can also check directly that the time-ordered exponential of \mathbf{A} is correctly determined from \mathbf{T} using either the general formula of Eq. (2.16) or, because the situation is so simple that all entries depend only on $t' - t$, we may use a Laplace transform with respect to $t' - t$. This gives $\mathbf{T}(s)$, and the inverse Laplace-transform of the resolvent $(\mathbf{I} - \mathbf{T}(s))_{11}^{-1}$ is the desired quantity. Both procedures give the same result, namely the derivative of $e^{\mathbf{A}t}$ as it should [26], i.e.,

$$(\mathbf{Id}_* - \mathbf{T})_{11}^{*-1}(t', 0) = \left(\sinh(2t') + \frac{1}{\sqrt{2}} \sinh(\sqrt{2}t') - \cosh(2t') \right) \Theta(t'),$$

which is indeed the derivative of Eq. (4.1).

Example 4.2 (Time-ordered exponential of a time-dependent matrix) In this example, we consider the 5×5 time-dependent matrix $\mathbf{A}(t', t) = \tilde{\mathbf{A}}(t')\Theta(t' - t)$ with

$$\tilde{\mathbf{A}}(t') = \begin{pmatrix} \cos(t') & 0 & 1 & 2 & 1 \\ 0 & \cos(t') - t' & 1 - 3t' & t' & 0 \\ 0 & t' & 2t' + \cos(t') & 0 & 0 \\ 0 & 1 & 2t' + 1 & t' + \cos(t') & t' \\ t' & -t' - 1 & -6t' - 1 & 1 - 2t' & \cos(t') - 2t' \end{pmatrix}.$$

The matrix $\tilde{\mathbf{A}}$ does not commute with itself at different times $\tilde{\mathbf{A}}(t')\tilde{\mathbf{A}}(t) - \tilde{\mathbf{A}}(t)\tilde{\mathbf{A}}(t') \neq 0$, and the corresponding differential system Eq. (1.1) has no known analytical solution. We use Algorithm 1 to determine the tridiagonal matching moment matrix \mathbf{T} such that $(\mathbf{A}^{*j})_{11} = (\mathbf{T}^{*j})_{11}$ for $j \in \mathbb{N}$. We define $\mathbf{w} := \mathbf{v}^H := (1, 0, 0, 0, 0)$, $\mathbf{w}_0 = \mathbf{w}1_*$, $\mathbf{v}_0 = \mathbf{v}1_*$, from which it follows that

$$\begin{aligned} \alpha_0(t', t) &= \cos(t')\Theta(t' - t), \\ \mathbf{w}_1 &= (0, 0, 1, 2, 1)\Theta(t' - t), \\ \widehat{\mathbf{v}}_1 &= (0, 0, 0, 0, t')^H \Theta(t' - t), \\ \beta_1(t', t) &= \frac{1}{2}(t'^2 - t^2)\Theta(t' - t). \end{aligned}$$

Observing that $\beta_1 = \Theta(t' - t) * t'\Theta(t' - t)$, we get $\beta_1^{*-1} = \frac{1}{t'}\delta'(t' - t) * \delta'(t' - t) = -\frac{1}{t'^2}\delta'(t' - t) + \frac{1}{t'}\delta''(t' - t)$, so that

$$\mathbf{v}_1 = \widehat{\mathbf{v}}_1 * \beta_1^{*-1} = (0, 0, 0, 0, 1)^H \delta'(t' - t),$$

which terminates the initialization phase of the Algorithm. We proceed with

$$\alpha_1(t', t) = \mathbf{w}_1 * \mathbf{A} * \mathbf{v}_1 = \cos(t)\Theta(t' - t),$$

$$\begin{aligned}
\mathbf{w}_2 &= \mathbf{w}_1 * \mathbf{A} - \alpha_1 * \mathbf{w}_1 - \beta_1 * \mathbf{w}_0, \\
&= (0, t' - t, t' - t, t' - t, 0)\Theta(t' - t), \\
\widehat{\mathbf{v}}_2 &= \mathbf{A} * \mathbf{v}_1 - \mathbf{v}_1 * \alpha_1 - \mathbf{v}_0 = (0, 0, 0, t, -2t)^H \delta(t' - t), \\
\beta_2 &= \mathbf{w}_2 * \mathbf{A} * \mathbf{v}_1 = t(t' - t)\Theta(t' - t).
\end{aligned}$$

As we did for β_1 , we factorize $\beta_2 = \Theta(t' - t) * t\Theta(t' - t)$ so that its $*$ -inverse is $\beta_2^{*-1} = \frac{1}{t'}\delta'(t' - t) * \delta'(t' - t) = \frac{1}{t'}\delta''$. Then

$$\mathbf{v}_2 = (0, 0, 0, 1, -2)^H \delta''(t' - t).$$

Continuing in this fashion yields the tridiagonal output matrix $\mathbf{T}_5 \equiv \mathbf{T}$,

$$\mathbf{T} = \begin{pmatrix} \cos(t')\Theta & \delta & 0 & 0 & 0 \\ \frac{1}{2}(t'^2 - t^2)\Theta & \cos(t)\Theta & \delta & 0 & 0 \\ 0 & t(t' - t)\Theta & \tilde{\alpha}_2(t', t)\Theta & \delta & 0 \\ 0 & 0 & -\frac{1}{2}(3t^2 - 4tt' + t'^2)\Theta & \tilde{\alpha}_3(t', t)\Theta & \delta \\ 0 & 0 & 0 & (-2t^2 + 3tt' - t'^2)\Theta & \tilde{\alpha}_4(t', t)\Theta \end{pmatrix},$$

with

$$\begin{aligned}
\tilde{\alpha}_2(t', t) &= (t' - t)\sin(t) + \cos(t), \\
\tilde{\alpha}_3(t', t) &= \frac{1}{2} \left(4(t' - t)\sin(t) - \left((t - t')^2 - 2 \right) \cos(t) \right), \\
\tilde{\alpha}_4(t', t) &= \frac{1}{6} \left(\left((t - t')^2 - 18 \right) (t - t')\sin(t) + \left(6 - 9(t - t')^2 \right) \cos(t) \right),
\end{aligned}$$

and the bases matrices

$$\mathbf{V}_5 = \begin{pmatrix} \delta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta^{(3)} & -2\delta^{(4)} \\ 0 & 0 & 0 & 0 & \delta^{(4)} \\ 0 & 0 & \delta'' & -\delta^{(3)} & \delta^{(4)} \\ 0 & \delta' & -2\delta'' & 2\delta^{(3)} & -3\delta^{(4)} \end{pmatrix}, \quad \mathbf{W}_5^H = \begin{pmatrix} \delta & 0 & 0 & 0 & 0 \\ 0 & 0 & \Theta & 2\Theta & \Theta \\ 0 & \Theta^{*2} & \Theta^{*2} & \Theta^{*2} & 0 \\ 0 & \Theta^{*3} & 2\Theta^{*3} & 0 & 0 \\ 0 & 0 & \Theta^{*4} & 0 & 0 \end{pmatrix}.$$

In all of these expressions, Θ and $\delta^{(n)}$ are short-hand notations respectively for $\Theta(t' - t)$ and $\delta^{(n)}(t' - t)$. All the required β_j^{*-1} were calculated using the strategies described in [18], getting the factorized $*$ -inverses

$$\beta_3^{*-1} = \frac{1}{t}\Theta(t' - t) * \delta^{(3)}(t' - t), \quad \beta_4^{*-1} = \frac{t'}{t^2}\Theta(t' - t) * \delta^{(3)}(t' - t).$$

We have also verified that $(\mathbf{A}^{*j})_{11} = (\mathbf{T}^{*j})_{11}$ holds for j up to 9. The $*$ -resolvent of \mathbf{T} has no closed-form expression, its Neumann series likely converging to a hitherto undefined special function. Ultimately, such difficulties are connected with the propensity of systems of coupled linear ordinary differential equations with non-constant coefficients to produce transcendent solutions.

5 Outlook: Numerical implementation

We do not expect closed-forms to exist in most cases for the entries of time-ordered matrix exponentials as these can involve complicated special functions [2]. Also, very large matrices $\mathbf{A}(t')$ are to be treatable by the algorithm for it to be relevant to most applications. For these reasons, it is fundamental to

implement the *-Lanczos algorithm numerically, e.g., using time discretization approximations.

As shown in [26], there exists an isometry Φ between the algebra of distributions of $D(I)$ equipped with the *-product and the algebra of *time-continuous* operators (for which the time variables t' and t serve as line and row indices). Consider, for simplicity, a discretization of the interval I with constant time step Δt ; then these operators become ordinary matrices. Specifically, given $f, g \in \text{Sm}_\Theta(I)$, their discretization counterparts are the *lower triangular* matrices F, G . Moreover, the function $f * g$ corresponds to the matrix $FG \Delta t$, with the usual matrix product. In other terms, the isometry Φ followed by a time discretization sends the *-product to the ordinary matrix product times Δt . Similarly, the Dirac delta distribution is sent to the identity matrix times $1/(\Delta t)$, the k th Dirac delta derivative $\delta^{(k)}$ is sent to the finite difference matrix

$$(M_{\delta^{(k)}})_{ij} = \frac{1}{(\Delta t)^{k+1}} \begin{cases} (-1)^{i-j} \binom{k}{i-j}, & \text{if } i \geq j \\ 0, & \text{else} \end{cases},$$

and Θ is sent to the matrix $(M_\Theta)_{ij} = 1$ if $i \geq j$ and 0 otherwise. Most importantly, in this picture, the *-inverse of a function $f(t', t) \in D(I)$ is given as $F^{-1}/(\Delta t)^2$, with F the lower triangular matrix corresponding to f . Moreover, the time-discretized version of the path-sum formulation Eq. (2.16) only involves ordinary matrix resolvents. At the same time, the final integration of $R_*(T_n)_{11}$ yielding $\mathbf{w}^H \mathbf{U} \mathbf{v}$ becomes a left multiplication by M_Θ . Therefore, a numerical implementation of the time-discretized *-Lanczos algorithm only requires *ordinary operations on triangular matrices*.

We can now meaningfully evaluate the numerical cost of the time-discretized version of the algorithm. Let N_t be the number of time subintervals in the discretization of I for both the t and t' time variables. Then time-discretized *-multiplications or *-inversions cost $O(N_t^3)$ operations. Considering a sparse time-dependent matrix $\mathbf{A}(t)$ with N_{nz} nonzero elements, the *-Lanczos algorithm therefore necessitates $O(N_i \times N_t^3 \times N_{nz})$ operations to obtain the desired $\mathbf{w}^H \mathbf{U} \mathbf{v}$. Here N_i is the number of iterations needed to get an error lower than a given tolerance. Unfortunately, as well-explained in [36], the presence of computational errors can slow down the (usual) Lanczos algorithm convergence. Hence, in general, we cannot assume $N_i \approx N$ since the *-Lanczos algorithm could analogously require more iterations. However, in many cases, the (usual) Lanczos algorithm demands *few* iterations to reach the tolerance also in finite precision arithmetic. We expect the *-Lanczos algorithm to behave analogously, giving $N_i \ll N$ in many cases. Concerning N_t , there is no reason to expect that it would depend on N since N_t controls the quality of individual generalized functions. We also remark that the *-Lanczos algorithm can exploit the sparsity structure of the matrix \mathbf{A} , making it inherently competitive when dealing with large sparse matrices that are typical of applications.

The classical numerical methods (e.g., Runge–Kutta methods) for the approximation of the system of ODEs (1.1) are known to perform poorly in certain cases. These include for example, very large system sizes, or in the presence of highly oscillatory coefficients. Consequently, in the last decades, novel techniques have been sought and proposed, many of which are based on the Magnus series; see, for instance, [4, 23, 54–61]. However, for large matrices, these methods are known to be highly consuming in resources. This motivates the research of novel approaches in particular for large-scale problems. Here the guaranteed convergence of the *-Lanczos algorithm in a finite number of iterations, the sequence of approximations it produces, its ability to exploit matrix sparsity and its relations with numerically well studied Lanczos procedures are all promising elements which justify further works on concrete numerical implementations. More precise theoretical results about the overall approximation quality and further issues on numerical applications of the present algorithm are beyond the scope of this work. They will appear together with a practical implementation of the algorithm in a future contribution.

6 Conclusion

In this work, we constructed the *-Lanczos algorithm as a biorthogonalization process of Krylov subspaces composed of distributions with respect to the *-product, a convolution-like product. The algorithm relies on a non-commutative operation and is analogous in spirit to the non-Hermitian Lanczos algorithm. The *-Lanczos algorithm can be used to express the element of a time-ordered exponential of size $N \times N$ by the path-sum continued fraction (2.16). To our knowledge, such an expression is the only one composed of $\mathcal{O}(N)$ scalar integro-differential equations. This approach generates controllable sequences of time-ordered exponential approximations, offers an innovative perspective of the connection between numerical linear algebra and differential calculus, and opens the door to efficient numerical algorithms for large scale computations.

Acknowledgments. We thank Francesca Arrigo, Des Higham, Jennifer Pestana, and Francesco Tudisco for their invitation to the University of Strathclyde, without which this work would not have come to fruition. The first author was supported in part by 2019 Alcohol project ANR-19-CE40-0006 and 2020 Magica project ANR-20-CE29-0007. The second author was supported by Charles University Research programs No. PRIMUS/21/SCI/009 and UNCE/SCI/023.

Declarations

- On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- [1] Dyson, F.J.: Divergence of perturbation theory in quantum electrodynamics. *Phys. Rev.* **85**(4), 631–632 (1952). <https://doi.org/10.1103/PhysRev.85.631>
- [2] Xie, Q., Hai, W.: Analytical results for a monochromatically driven two-level system. *Phys. Rev. A* **82**, 032117 (2010)
- [3] Hortaçsu, M.: Heun functions and some of their applications in physics. *Adv. High Energy Phys.* **2018**, 8621573 (2018)
- [4] Blanes, S., Casas, F., Oteo, J.A., Ros, J.: The Magnus expansion and some of its applications. *Phys. Rep.* **470**(5), 151–238 (2009)
- [5] Autler, S.H., Townes, C.H.: Stark effect in rapidly varying fields. *Phys. Rev.* **100**, 703–722 (1955)
- [6] Shirley, J.H.: Solution of the Schrödinger equation with a Hamiltonian periodic in time. *Phys. Rev.* **138**, 979–987 (1965)
- [7] Lauder, M.A., Knight, P.L., Greenland, P.T.: Pulse-shape effects in intense-field laser excitation of atoms. *Opt. Acta* **33**(10), 1231–1252 (1986)
- [8] Reid, W.T.: Riccati matrix differential equations and non-oscillation criteria for associated linear differential systems. *Pacific J. Math.* **13**(2), 665–685 (1963)
- [9] Kwakernaak, H., Sivan, R.: *Linear Optimal Control Systems* vol. 1. Wiley-interscience, New York (1972)
- [10] Corless, M., Frazho, A.: *Linear Systems and Control: an Operator Perspective*. Pure and Applied Mathematics. Marcel Dekker, New York (2003)
- [11] Blanes, S.: High order structure preserving explicit methods for solving linear-quadratic optimal control problems. *Numer. Algorithms* **69**(2), 271–290 (2015)
- [12] Benner, P., Cohen, A., Ohlberger, M., Willcox, K.: *Model Reduction and Approximation: Theory and Algorithms*. Computational Science and Engineering. SIAM, Philadelphia (2017)
- [13] Kučera, V.: A review of the matrix Riccati equation. *Kybernetika* **9**(1), 42–61 (1973)
- [14] Abou-Kandil, H., Freiling, G., Ionescu, V., Jank, G.: *Matrix Riccati*

Equations in Control and Systems Theory. Systems & Control: Foundations & Applications. Birkhäuser, Basel (2003)

- [15] Hached, M., Jbilou, K.: Numerical solutions to large-scale differential Lyapunov matrix equations. *Numer. Algorithms* **79**(3), 741–757 (2018)
- [16] Kirsten, G., Simoncini, V.: Order reduction methods for solving large-scale differential matrix Riccati equations. *SIAM J. Sci. Comput.* **42**(4), 2182–2205 (2020). <https://doi.org/10.1137/19m1264217>
- [17] Giscard, P.-L., Pozza, S.: Tridiagonalization of systems of coupled linear differential equations with variable coefficients by a Lanczos-like method. *Linear Algebra and its Applications* **624**, 153–173 (2021). <https://doi.org/10.1016/j.laa.2021.04.011>
- [18] Giscard, P.-L., Pozza, S.: Lanczos-like algorithm for the time-ordered exponential: The *-inverse problem. *Appl. Math.* **65**(6), 807–827 (2020). <https://doi.org/10.21136/am.2020.0342-19>
- [19] Magnus, W.: On the exponential solution of differential equations for a linear operator. *Comm. Pure Appl. Math.* **7**(4), 649–673 (1954)
- [20] Casas, F.: Sufficient conditions for the convergence of the Magnus expansion. *J. Phys. A* **40**(50), 15001–15017 (2007). <https://doi.org/10.1088/1751-8113/40/50/006>
- [21] Fel'dman, E.B.: On the convergence of the Magnus expansion for spin systems in periodic magnetic fields. *Phys. Lett.* **104A**(9), 479–481 (1984)
- [22] Maricq, M.M.: Convergence of the Magnus expansion for time dependent two level systems. *J. Chem. Phys.* **86**(10), 5647–5651 (1987)
- [23] Iserles, A., Munthe-Kaas, H.Z., Nørsett, S.P., Zanna, A.: Lie-group methods. *Acta Numer.* **9**, 215–365 (2000). <https://doi.org/10.1017/S0962492900002154>
- [24] Moan, P.C., Niesen, J.: Convergence of the Magnus series. *Found. Comput. Math.* **8**(3), 291–301 (2007). <https://doi.org/10.1007/s10208-007-9010-0>
- [25] Sánchez, S., Casas, F., Fernández, A.: New analytic approximations based on the Magnus expansion. *J. Math. Chem.* **49**(8), 1741–1758 (2011)
- [26] Giscard, P.-L., Lui, K., Thwaite, S.J., Jaksch, D.: An exact formulation of the time-ordered exponential using path-sums. *J. Math. Phys.* **56**(5), 053503 (2015)

- [27] Balasubramanian, V., DeCross, M., Kar, A., Parrikar, O.: Quantum complexity of time evolution with chaotic Hamiltonians. *J. High Energ. Phys.* **134** (2020)
- [28] Giscard, P.-L., Bonhomme, C.: Dynamics of quantum systems driven by time-varying Hamiltonians: Solution for the Bloch-Siegert Hamiltonian and applications to NMR. *Phys. Rev. Research* **2**, 023081 (2020). <https://doi.org/10.1103/PhysRevResearch.2.023081>
- [29] J. Flum, M.G.: The parameterized complexity of counting problems. *SIAM J. Comput.* **33**, 892–922 (2004)
- [30] Moler, C., Van Loan, C.: Nineteen dubious ways to compute the exponential of a matrix. *SIAM Rev.* **20**(4), 801–836 (1978)
- [31] Moler, C., Van Loan, C.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.* **45**(1), 3–49 (2003)
- [32] Higham, N.J.: *Functions of Matrices. Theory and Computation*, p. 425. SIAM, Philadelphia (2008)
- [33] Gutknecht, M.H.: A completed theory of the unsymmetric Lanczos process and related algorithms. I. *SIAM J. Matrix Anal. Appl.* **13**(2), 594–639 (1992)
- [34] Gutknecht, M.H.: A completed theory of the unsymmetric Lanczos process and related algorithms. II. *SIAM J. Matrix Anal. Appl.* **15**(1), 15–58 (1994)
- [35] Golub, G.H., Meurant, G.: *Matrices, Moments and Quadrature with Applications*. Princeton Ser. Appl. Math., p. 363. Princeton University Press, Princeton (2010)
- [36] Liesen, J., Strakoš, Z.: *Krylov Subspace Methods: Principles and Analysis*. Numer. Math. Sci. Comput. Oxford University Press, Oxford (2013)
- [37] Pozza, S., Pranić, M.S., Strakoš, Z.: Gauss quadrature for quasi-definite linear functionals. *IMA J. Numer. Anal.* **37**(3), 1468–1495 (2017)
- [38] Pozza, S., Pranić, M.S., Strakoš, Z.: The Lanczos algorithm and complex Gauss quadrature. *Electron. Trans. Numer. Anal.* **50**, 1–19 (2018)
- [39] Estrada, E., Rodríguez-Velázquez, J.A.: Subgraph centrality in complex networks. *Phys. Rev. E* **71**, 056103 (2005)
- [40] Benzi, M., Boito, P.: Quadrature rule-based bounds for functions of adjacency matrices. *Linear Algebra Appl.* **433**(3), 637–652 (2010)

- [41] Parlett, B.N.: Reduction to tridiagonal form and minimal realizations. *SIAM J. Matrix Anal. Appl.* **13**(2), 567–593 (1992)
- [42] Draux, A.: *Polynômes Orthogonaux Formels*. Lecture Notes in Math., vol. 974, p. 625. Springer, Berlin (1983)
- [43] Halperin, I., Schwartz, L.: *Introduction to the Theory of Distributions*. University of Toronto Press, Toronto (19 Feb. 2019). <https://doi.org/10.3138/9781442615151>. <https://toronto.degruyter.com/view/title/550976>
- [44] Schwartz, L.: *Théorie Des Distributions*, Nouvelle édition, entièrement corrigée, refondue et augmentée edn. Hermann, Paris (1978)
- [45] Volterra, V., Pèrès, J.: *Leçons sur la Composition et les Fonctions Permutables*. Éditions Jacques Gabay, Paris (1928)
- [46] Giscard, P.-L., Thwaite, S.J., Jaksch, D.: Walk-sums, continued fractions and unique factorisation on digraphs. arXiv:1202.5523 [cs.DM] (2012)
- [47] Kılıç, E.: Explicit formula for the inverse of a tridiagonal matrix by backward continued fractions. *Appl. Math. Comput.* **197**(1), 345–357 (2008)
- [48] Taylor, D.R.: *Analysis of the look ahead Lanczos algorithm*. PhD thesis, University of California, Berkeley (1982)
- [49] Parlett, B.N., Taylor, D.R., Liu, Z.A.: A look-ahead Lanczos algorithm for unsymmetric matrices. *Math. Comp.* **44**(169), 105–124 (1985)
- [50] Freund, R.W., Gutknecht, M.H., Nachtigal, N.M.: An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM J. Sci. Comput.* **14**, 137–158 (1993)
- [51] Brezinski, C., Redivo Zaglia, M., Sadok, H.: Avoiding breakdown and near-breakdown in Lanczos type algorithms. *Numer. Algorithms* **1**(3), 261–284 (1991)
- [52] Brezinski, C., Redivo Zaglia, M., Sadok, H.: A breakdown-free Lanczos type algorithm for solving linear systems. *Numer. Math.* **63**(1), 29–38 (1992)
- [53] Pozza, S., Pranić, M.: The Gauss quadrature for general linear functionals, Lanczos algorithm, and minimal partial realization. *Numer. Algor* **88**, 647–678 (2021). <https://doi.org/10.1007/s11075-020-01052-y>
- [54] Hochbruck, M., Lubich, C.: Exponential integrators for quantum-classical molecular dynamics. *BIT Numer. Math.* **39**(4), 620–645 (1999). <https://doi.org/10.1023/A:1022335122807>

- [55] Budd, C.J., Iserles, A., Iserles, A., Nørsett, S.P.: On the solution of linear differential equations in lie groups. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **357**(1754), 983–1019 (1999) <https://arxiv.org/abs/https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.1999.0362>. <https://doi.org/10.1098/rsta.1999.0362>
- [56] Iserles, A.: On the global error of discretization methods for highly-oscillatory ordinary differential equations. *BIT Numer. Math.* **42**(3), 561–599 (2002). <https://doi.org/10.1023/A:1022049814688>
- [57] Iserles, A.: On the method of Neumann series for highly oscillatory equations. *BIT Numer. Math.* **44**(3), 473–488 (2004). <https://doi.org/10.1023/B:BITN.0000046810.25353.95>
- [58] Degani, I., Schiff, J.: RCMS: Right correction Magnus series approach for oscillatory ODEs. *J. Comput. Appl. Math.* **193**(2), 413–436 (2006). <https://doi.org/10.1016/j.cam.2005.07.001>
- [59] Cohen, D., Jahnke, T., Lorenz, K., Lubich, C.: Numerical integrators for highly oscillatory Hamiltonian systems: A review. In: Mielke, A. (ed.) *Analysis, Modeling and Simulation of Multiscale Problems*, pp. 553–576. Springer, Berlin, Heidelberg (2006)
- [60] Bader, P., Iserles, A., Kropielnicka, K., Singh, P.: Efficient methods for linear Schrödinger equation in the semiclassical regime with time-dependent potential. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **472**, 20150733 (2016)
- [61] Blanes, S., Casas, F.: *A Concise Introduction to Geometric Numerical Integration*. CRC Press, Boca Raton, FL (2017)