Lanczos-like method for the time-ordered exponential

Pierre-Louis Giscard[1] Stefano Pozza[2]

[1]Université du Littoral Côte d'Opale, Calais

[2]Charles University, Prague

June 25, 2019

Time-ordered exponential

Let $t \in I \subseteq \mathbb{R}$, $\mathsf{A}(t)$ a time dependent matrix. The time-ordered exponential is the unique solution $\mathsf{U}(t)$ of the system of coupled linear differential equations

$$\mathsf{A}(t)\mathsf{U}(t) = \frac{d}{dt}\mathsf{U}(t), \quad \mathsf{U}(0) = \mathsf{I}.$$

If $\mathsf{A}(t')\mathsf{A}(t) = \mathsf{A}(t)\mathsf{A}(t')$ for all t, t', then

$$\mathsf{U}(t) = \exp\left(\int_0^t \mathsf{A}(\tau)\, d\tau\right).$$

Otherwise, U has no known explicit form in terms of A. It can be denoted by the time-ordering operator $\mathcal{T}$ as

$$\mathsf{U}(t) = \mathcal{T}\exp\left(\int_0^t \mathsf{A}(\tau)\, d\tau\right).$$

[Dyson, 1952]

Time-ordered exponential

The time-ordering expression is more a notation than an explicit form as the action of the time-ordering operator is very difficult to evaluate.

- Applications: system dynamics (quantum physics); see, e.g., [Blanes & al., 2009].
- Classical approaches Magnus series [Magnus, 1954]: infinite series of increasing complexity (almost always divergent). Using Floquet theory: only A periodic, the solution is given in terms of a infinite system. See, e.g., [Blanes & al., 2009].
- Path-sum approach: based on the *path-sum* decomposition of the graph whose A is the adjacency matrix [Giscard & al., 2015]. The expression has a finite number of terms, but it may be too complicate.

Outline

- Idea: model reduction + path-sum approach;
- Lanczos algorithm properties (time-independent matrices);
- $*$-Lanczos algorithm for time-ordered exponentials.

Warning 1: Work in progress!

Comments and suggestions are more than welcome!

Warninig 2: No computational approach

We will derive an expression for the solution on the base of which a further numerical approach may be developed.

Strategy

Let $t' \geq t$ and let $\mathsf{A}_1(t', t)$ and $\mathsf{A}_2(t', t)$ be (doubly) time-dependent matrices. We define the convolution-like product

$$(\mathsf{A}_2 * \mathsf{A}_1)(t', t) := \int_t^{t'} \mathsf{A}_2(t', \tau)\mathsf{A}_1(\tau, t)\, \mathrm{d}\tau.$$

Consider the $*$-resolvent

$$\mathsf{R}_*(\mathsf{A}) := (1_* - \mathsf{A})^{*-1} = \mathsf{I}1_* + \sum_{k \geq 1} \mathsf{A}^{*k}$$

(it exists if every entry of A is bounded for all t', t). Then

$$\mathsf{U}(t', t) = \int_t^{t'} \mathsf{R}_*(\mathsf{A})(\tau, t)\, d\tau,$$

with $1_* = \delta(t' - t)$ the Dirac delta function; [Giscard & al., 2015].

Path-sum formulation

If we look at A as the adjacency matrix of the graph G with time-dependent weights, then the (i, j) element of the $*$-resolvent, $(R_*)_{ij}$, can be rewritten in terms of the simple paths from i to j in G; [Giscard & al., 2015].

This approach provides exact solutions and it is unconditionally convergent. However, it suffers from drawbacks:

- It requires to find all the simple cycles and simple paths of the graph G (computationally expensive);
- The expression for U may be too complicated due to the structure of the graph G.

Idea

We want to compute a matrix T so that

- T is tridiagonal, i.e, it correspond to a graph which is a path (with self-loops);
- $R_*(\mathrm{T})_{11} = R_*(\mathrm{A})_{ij}$, i.e., they have the same $*$-moments

$$(T^{*k})_{11} = (A^{*k})_{ij}, \quad j = 0, 1, 2, \dots .$$

In the simple case in which A is time-independent, a T with analogous properties is give as the output of the last iteration of Lanczos algorithm.

We will derive a Lanczos-like method based on an $*$-orthogonalization process.

Non-Hermitian Lanczos algorithm

Let A be a time-independent matrix and $\mathbf{v}, \mathbf{w}$ time-independent vectors. Our goal is to approximate

$$\mathbf{w}^H \exp(\mathrm{A})\,\mathbf{v}.$$

Consider the Krylov subspaces

$$\mathrm{span}\{\mathbf{v}, \mathrm{A}\,\mathbf{v}, \dots, \mathrm{A}^{n-1}\,\mathbf{v}\}, \quad \mathrm{span}\{\mathbf{w}, \mathrm{A}^H\mathbf{w}, \dots, (\mathrm{A}^H)^{n-1}\mathbf{w}\}.$$

Assuming not breakdowns, the non-Hermitian Lanczos computes the matrices

$$\mathrm{V}_n = [\mathbf{v}_0, \dots, \mathbf{v}_{n-1}], \quad \mathrm{W}_n = [\mathbf{w}_0, \dots, \mathbf{w}_{n-1}]$$

respectively basis of the Krylov subspaces, so that $\mathrm{W}_n^H \mathrm{V}_n = I$.

Non-Hermitian Lanczos algorithm

The tridiagonal matrix defined as

$$\mathrm{T}_n = \mathrm{W}_n^H \mathrm{A} \mathrm{V}_n,$$

satisfies the matching moment property

$$\mathbf{w}^H \mathrm{A}^k \mathbf{v} = \mathbf{e}_1^H \mathrm{T}_n^k \mathbf{e}_1, \quad k = 0, \dots, 2n-1.$$

Hence we get the approximation (model reduction)

$$\mathbf{w}^H \exp(\mathrm{A})\, \mathbf{v} \approx \mathbf{e}_1^H \exp(\mathrm{T}_n)\, \mathbf{e}_1;$$

e.g., [Golub, Meurant, 2010].
In particular, for n equal to the size of A (or for lucky or incurable breakdowns) we get exactness.

$*$-resolvent approximation

Now, $A(t)$ is time-dependent and we want to approximate the i, j element of the time-ordered exponential of A, i.e.,

$$U(t', t)_{i,j} = \mathbf{e}_i^H \mathcal{T} \exp\left(\int_t^{t'} A(\tau)\, d\tau \right) \mathbf{e}_j = \int_t^{t'} \mathbf{e}_i^H R_*(A)(\tau, t) \mathbf{e}_j \, d\tau.$$

Hence we want to approximate the value

$$\mathbf{w}^H R_*(A)(t', t)\, \mathbf{v},$$

with $\mathbf{v}, \mathbf{w}$ time-independent vectors.

Time-dependent Krylov subspaces

Given the distributions $\alpha_0(t', t), \ldots, \alpha_k(t', t)$ (coefficients),

$$\text{matrix } *\text{-polynomial:} \quad p(\mathsf{A})(t', t) := \sum_{j=0}^{k} \left(\mathsf{A}^{*j} * \alpha_j\right)(t', t),$$

$$\text{dual } *\text{-polynomial:} \quad p^D(\mathsf{A})(t', t) := \sum_{j=0}^{k} \left(\overline{\alpha}_j * (\mathsf{A}^{*j})^H\right)(t', t),$$

and hence the time-dependent Krylov subspaces as

$$\mathcal{K}_n(\mathsf{A}, \mathbf{v})(t', t) := \{ \, (\, p(\mathsf{A})\mathbf{v}) \mid p \text{ of degree } \leq n-1\} \, ,$$
$$\mathcal{K}_n^D(\mathsf{A}^H, \mathbf{w})(t', t) := \left\{ \left(\mathbf{w}^H p^D(\mathsf{A}^H) \right) \mid p \text{ of degree } \leq n-1 \right\} .$$

Building up the $*$-Lanczos process

We aim to build basis for $\mathcal{K}_n(\mathsf{A}, \mathbf{v})$ and $\mathcal{K}_n^D(\mathsf{A}^H, \mathbf{w})$

$$\mathsf{V}_n = [\mathbf{v}_0, \dots, \mathbf{v}_{n-1}], \quad \mathsf{W}_n^H = [\mathbf{w}_0, \dots, \mathbf{w}_{n-1}]^H$$

which are $*$-biorthonormal, i.e.,

$$\mathsf{W}_n^H * \mathsf{V}_n = \mathsf{I}1_*.$$

Assuming $\mathbf{w}^H\mathbf{v} = 1$, the first vectors are

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

Building up the $*$-Lanczos process

Consider a vector $\widehat{\boldsymbol{v}}_1 \in \mathcal{K}_2(\mathsf{A}, \boldsymbol{v})$

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

Under the condition $\boldsymbol{w}_0^H * \widehat{\boldsymbol{v}}_1 = 0$, we get

$$\alpha_0 = \boldsymbol{w}_0^H * \mathsf{A} * \boldsymbol{v}_0 = \boldsymbol{w}^H \mathsf{A}\, \boldsymbol{v} \tag{1}$$

Similarly,

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

with α_0 given by (1).

Building up the $*$-Lanczos process

The condition

$$\boldsymbol{w}_1^H * \boldsymbol{v}_1 = 1_*,$$

is satisfied defining

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

with

$$\beta_1 = \widehat{\boldsymbol{w}}_1^H * \widehat{\boldsymbol{v}}_1 = \boldsymbol{w}^H \mathsf{A}^{*2} \, \boldsymbol{v} - \alpha_0^{*2}.$$

We assume β_1 $*$-invertible and computable. We will discuss this issue later.

Building up the $*$-Lanczos process

By induction, consider the vector

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

The condition $\boldsymbol{w}_j^H * \widehat{\boldsymbol{v}}_n = 0$ for $j = 0, \dots, n-1$ gives

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

Since $\boldsymbol{w}_j^H * \mathsf{A} \in \mathcal{K}_{j+1}^D(\mathsf{A}^H, \boldsymbol{w})$ we get

$$\gamma_j = 0, \quad j = 0, \dots, n-3.$$

Building up the $*$-Lanczos process

Assuming $\boldsymbol{v}_{-1} = \boldsymbol{w}_{-1} = 0$, we get the three-term recurrences

$$\boldsymbol{w}_n^H = \boldsymbol{w}_{n-1}^H * \mathsf{A} - \alpha_{n-1} * \boldsymbol{w}_{n-1}^H - \beta_{n-1} * \boldsymbol{w}_{n-2}^H,$$
$$\boldsymbol{v}_n * \beta_n = \mathsf{A} * \boldsymbol{v}_{n-1} - \boldsymbol{v}_{n-1} * \alpha_{n-1} - \boldsymbol{v}_{n-2},$$

with the coefficients given by

$$\alpha_{n-1} = \boldsymbol{w}_{n-1}^H * \mathsf{A} * \boldsymbol{v}_{n-1}, \quad \beta_n = \boldsymbol{w}_n^H * \mathsf{A} * \boldsymbol{v}_{n-1}.$$

- If β_n is not $*$-invertible we have a breakdown;
- We assume no breakdowns in this first study.

Building up the $*$-Lanczos process

For $n \geq 2$, setting $\gamma_n = (\boldsymbol{w}_n^H + \boldsymbol{w}_{n-2}^H) * \mathrm{A} * \boldsymbol{v}_{n-1}$

$$\beta_n = \gamma_n - 1_*, \quad \beta_n^{*-1} = R_*(\gamma_n),$$

which can be reformulated as a Volterra equations of the second kind

$$R_* = 1_* + \gamma_n * R_*.$$

If γ_n is bounded over I^2, then R_* necessarily exists and the Neumann series $\sum_{k\geq 0} \gamma_n^{*k}$ converges to it super-geometrically.

$*$-inverse, one time

Let $a(t') \neq 0$ be a function of a single time variable $t' \in I$. Then a^{*-1} exists as a distribution and its given by

$$a^{*-1}(t', t) = \frac{d}{dt'}\left[\frac{\delta(t'-t)}{a(t')}\right] = \frac{\delta'(t'-t)}{a(t')} - \frac{a'(t')}{a(t')^2}\delta(t'-t).$$

δ' and a' are the derivatives of the Dirac delta distribution and a, respectively. Now let $f(t', t)$ be a distribution. Then

$$\left(a^{*-1} * f\right)(t', t) = \frac{d}{dt'}\left[\frac{f(t', t)}{a(t')}\right]$$

$$\left(f * a^{*-1}\right)(t', t) = -\frac{1}{a(t)}\frac{d}{dt}\left[f(t', t)\right].$$

$*$-Lanczos Algorithm

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

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

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

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

$\beta_1 = \boldsymbol{w}^H \mathsf{A}^{*2}\, \boldsymbol{v} - \alpha_0^{*2},$

Determine β_1^{*-1},

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

For $n = 2, \dots$

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

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

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

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

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

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

end.

Tridiagonal matrix

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

and recall $\mathsf{V}_n := [\boldsymbol{v}_0, \dots, \boldsymbol{v}_{n-1}]$ and $\mathsf{W}_n := [\boldsymbol{w}_0, \dots, \boldsymbol{w}_{n-1}]$. Then

$$\mathsf{A} * \mathsf{V}_n = \mathsf{V}_n * \mathsf{T}_n + (\boldsymbol{v}_n * \beta_n)\boldsymbol{e}_n^H$$
$$\mathsf{W}_n^H * \mathsf{A} = \mathsf{T}_n * \mathsf{W}_n^H + \boldsymbol{e}_n \boldsymbol{w}_n^H.$$

Hence T_n is the $*$-projection of A onto $\mathcal{K}_n(\mathsf{A}, \boldsymbol{v})$ along the direction of $\mathcal{K}_n^D(\mathsf{A}^H, \boldsymbol{w})$, i.e.,

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

Matching moment

Matching moment property

Let A, $\boldsymbol{w}$, $\boldsymbol{v}$ and T_n be as described above, then

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

In particular, for n equal to the size of A we get

$$\boldsymbol{w}^H (\mathrm{A}^{*k})\, \boldsymbol{v} = \boldsymbol{e}_1^H (\mathrm{T}_n^{*k})\, \boldsymbol{e}_1, \quad \text{for} \quad k = 0, 2, \dots,$$

and, hence,

$$\boldsymbol{w}^H R_*(\mathrm{A}) \boldsymbol{v} = \boldsymbol{e}_1^H R_*(\mathrm{T}_n) \boldsymbol{e}_1.$$

Idea of the proof

Inspired by the classical Lanczos algorithm, we follow the analogy with formal orthogonal polynomial properties; see [Draux, 1982], also the surveys [Pozza & al., 2017, 2018].
Define the set of $*$-polynomials

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

with coefficients the $\gamma_j(t', t)$ distributions.
Consider a $*$-sesquilinear form $[\cdot, \cdot] : \mathcal{P}_* \times \mathcal{P}_* \to \mathbb{C}[I^2]$

$$[q * \alpha, p * \beta] = \overline{\alpha} * [q, p] * \beta,$$
$$[q_1 + q_2, p_1 + p_2] = [q_1, p_1] + [q_2, p_1] + [q_1, p_2] + [q_2, p_2].$$

Idea of the proof

The polynomials $p_0, p_1, \dots$ and $q_0, q_1, \dots$ are $*$-biorthonormal with respect to $[\cdot,\cdot]$, if $[q_i, p_j] = \delta_{ij} 1_*$.

Lemma

Let $p_0, \dots, p_{n-1}$ and $q_0, \dots, q_{n-1}$ be $*$-biorthonormal with respect to $[\cdot,\cdot]_A$ and to $[\cdot,\cdot]_B$. If $[1_*, 1_*]_A = [1_*, 1_*]_B = 1_*$, then

$$[\lambda^{*k}, 1_*]_A = [\lambda^{*k}, 1_*]_B, \quad k = 0, \dots, 2n-1.$$

Finally we showed that the forms

$$[q,p] := \boldsymbol{w}^H \, \overline{q}^D(\mathrm{A}^H) * p(\mathrm{A}) \, \boldsymbol{v}., \quad [q,p]_n := \boldsymbol{e}_1^H \, \overline{q}^D(\mathrm{T}_n^H) * p(\mathrm{T}_n) \, \boldsymbol{e}_1,$$

have the same first n $*$-biorthonormal polynomials.

$*$-Lanczos approximation

Consider the time-ordered exponential U_n given by

$$\mathrm{T_n}(t)\mathrm{U_n}(t) = \frac{d}{dt}\mathrm{U}_n(t).$$

The matching moment property justify the use of the approximation

$$\boldsymbol{w}^H \mathrm{U}(t)\boldsymbol{v} \approx \boldsymbol{e}_1^H \mathrm{U}_n(t)\,\boldsymbol{e}_1 = \boldsymbol{e}_1^H \int_0^t R_*(\mathrm{T}_n)(\tau,0)\,d\tau\,\boldsymbol{e}_1.$$

The path-sum method gives the finite $*$-continued fraction

$$\Big(R_*(\mathrm{T}_n)(\tau,0)\Big)_{11} = \Big(1_* - \alpha_0 - \big(1_* - \alpha_1 - (\cdots * \beta_{n-1})^{*-1} \cdots * \beta_2\big)^{*-1} * \beta_1\Big)^{*-1},$$

see [Giscard & al. 2012, 2015].

$*$-Lanczos approximation

- The method converges at most in N iteration, N being the size of A;
- The intermediate approximations U_n have a clear meaning in terms of moment matching;
- An explicit expression for β_n^{*-1} may not be easily computed (work in progress).

Error bound?

We have not yet derived an a priori bound for the error

$$E_n(t) = \int_0^t \sum_{k=2n}^{\infty} \mathbf{w}^H \mathrm{A}^{*k}(\tau)\,\mathbf{v} - \mathbf{e}_1^H \mathrm{T}_n^{*k}(\tau,0)\,\mathbf{e}_1 \mathrm{d}\tau.$$

However, notice that if we it is possible to bound

$$\sup_{t\in I} |\mathbf{w}^H \mathrm{A}(t)\,\mathbf{v}| \leq C, \quad \sup_{t\in I} |\mathbf{e}_1^H \mathrm{T}_n(t,0)\,\mathbf{e}_1| \leq D,$$

then we get

$$|E_n(t)| \leq \frac{((C+D)t)^{2n+1} e^{(C+D)t}}{(2n+1)!}.$$

Note that

$$\left| \int_0^t A^{*2n}(\tau) d\tau \right| \leq C^{*2n} \int_0^t d\tau = \frac{(Ct)^{2n+1}}{(2n+1)!}.$$

Breakdown

When β_n is not invertible we have a breakdown. We distinguish two cases

- Lucky breakdown: if either $\mathbf{w}_n \equiv 0$ or $\mathbf{v}_n \equiv 0$;
- Serious breakdown: if $\mathbf{v}_n, \mathbf{w}_n \neq 0$, and $\mathbf{w}_n^H * \mathbf{v}_n$ is not invertible.

Proposition

If $\mathbf{v}_n = 0$ (or $\mathbf{w}_n = 0$), then

$$\boldsymbol{w}^H(\mathrm{A}^{*k})\,\boldsymbol{v} = \boldsymbol{e}_1^H(\mathrm{T}_n^{*k})\,\boldsymbol{e}_1, \quad \text{for} \quad k = 0, 1, \ldots .$$

Hence

$$\boldsymbol{w}^H \mathrm{U}(t)\boldsymbol{v} = \boldsymbol{e}_1^H \int_0^t \left(1_* - \mathrm{T}_n\right)^{*-1}(\tau, 0)\, d\tau\, \boldsymbol{e}_1.$$

Serious breakdown

- We do not treat the possible solutions to a general serious breakdowns;
- A possible way to deal with it is to use a look-ahead strategy analogous to what can be done for the non-Hermitian Lanczos algorithm.

A particular and relevant case of serious breakdown.

If A is a sparse non-Hermitian matrix, then it may be that $A_{ij} = 0$ and ${A^{*2}}_{ij} = 0$. Then $*$-Lanczos with inputs $A, \mathbf{e}_i, \mathbf{e}_j$ gives $\beta_1 = 0$. To fix this issue, we can consider the problem

$$\mathbf{e}_i^H U(t)\mathbf{e}_j = (\mathbf{e} + \mathbf{e}_i)^H U(t)\mathbf{e}_j - \mathbf{e}^H U(t)\mathbf{e}_j,$$

with $\mathbf{e} = (1, \dots, 1)^H$. Then one can approximate the $(\mathbf{e} + \mathbf{e}_i)^H U \mathbf{e}_j$ and $\mathbf{e}^H U \mathbf{e}_j$ separately, which now are less likely going to have a breakdown at the first step since $\mathbf{e}$ is full; e.g., [Golub, Meurant, 2010].

Example: Ordinary matrix exponential

Consider a constant matrix

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

Since A commutes with itself at all times, $\mathcal{T} e^{\int \mathsf{A}(\tau) d\tau} \equiv e^{\mathsf{A}(t)}$.

$$\left(e^{\mathsf{A}(t)}\right)_{11} = -\frac{1}{2}\sinh(2t) + \frac{1}{2}\cosh(2t) + \frac{1}{2}\cosh\left(\sqrt{2}t\right).$$

Note that we chose a symmetric matrix for simplicity.

Example: Ordinary matrix exponential

We compute the result by $*$-Lanczos. We set

$$\boldsymbol{w}^H := \boldsymbol{v}^H := (1, 0, 0), \quad \boldsymbol{w}_0 = \boldsymbol{w}1_*, \quad \boldsymbol{v}_0 = \boldsymbol{v}1_*.$$

Note that in such a simple setting we have

$$\mathsf{A}^{*n} = \mathsf{A}^n \times 1^{*n} = \mathsf{A} \times (t' - t)^{n-1}/(n-1)!.$$

It follows

$$\alpha_0 = \boldsymbol{w}^H \mathsf{A}\, \boldsymbol{v} = -1,$$

$$\boldsymbol{w}_1^H = \boldsymbol{w}^H \mathsf{A} - \alpha_0\, \boldsymbol{w}^H = \widehat{\boldsymbol{v}}_1^H = (0, 1, 1),$$

$$\beta_1 = \boldsymbol{w}^H \mathsf{A}^2 \boldsymbol{v} \times (t' - t) - \alpha_0^2 (t' - t) = 2(t' - t),$$

$$\beta_1^{*-1} = \frac{1}{2}\delta''(t' - t).$$

Example: Ordinary matrix exponential

We rescaled $\widehat{\boldsymbol{v}}_1$

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

from which we obtain

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

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

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

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

$\Theta(\cdot)$ is the Heaviside theta function ($\Theta(0)=1$), recall that $t' \geq t$.

Example: Ordinary matrix exponential

β_2 is $*$-invertible with

$$\beta_2^{*-1} = 4\delta''(t' - t),$$

and we get

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

Example: Ordinary matrix exponential

We have determined the $*$-Lanczos matrices T, V and W.

$$\mathsf{T}_3 = \begin{pmatrix} -1 & \delta(t'-t) & 0 \\ 2(t'-t) & \frac{1}{2} & \delta(t'-t) \\ 0 & \frac{1}{4}(t'-t) & -\frac{3}{2} \end{pmatrix},$$

$$\mathsf{V}_3 = \begin{pmatrix} \delta(t'-t) & 0 & 0 \\ 0 & \frac{1}{2}\delta'(t'-t) & \delta''(t'-t) \\ 0 & \frac{1}{2}\delta'(t'-t) & -\delta''(t'-t) \end{pmatrix},$$

$$\mathsf{W}_3 = \begin{pmatrix} \delta(t'-t) & 0 & 0 \\ 0 & 1 & 1 \\ 0 & \frac{1}{2}(t'-t) & \frac{1}{2}(t'-t) \end{pmatrix}.$$

Example: Ordinary matrix exponential

Using the continued fraction formulation we get

$$\left(1_* + 1 - \left(1_* - \frac{1}{2} - \left(1_* + \frac{3}{2}\right)^{*-1} * \frac{1}{4}(t'-t)\right)^{*-1} * 2(t'-t)\right)^{*-1}.$$

Since all the entries depend only on $t'-t$, we may also use a Laplace transform approach.

$$(\mathsf{Id}_* - \mathsf{T})^{*-1}_{11}(t,0) = \sinh(2t) + \frac{1}{\sqrt{2}}\sinh\left(\sqrt{2}t\right) - \cosh(2t),$$

which is indeed the derivative of

$$\left(e^{\mathsf{A}(t)}\right)_{11} = -\frac{1}{2}\sinh(2t) + \frac{1}{2}\cosh(2t) + \frac{1}{2}\cosh\left(\sqrt{2}t\right).$$

Conclusion

$*$-Lanczos is able to express the solution of a coupled linear differential equations with non-constant coefficient.
It presents several difficulties ($*$-inverses, integral solutions, ...).
Nevertheless,

- It is convergent in a finite number of steps;
- It express the solution with a finite formulation (as the path-sum method);
- It admits an expression in terms of a simple finite continued fraction.

Such a general approach has been so far out of reach.

Future developments and open problems

- Convergence properties of the approximation;
- Explicit formulation of the $*$-inverses β_n^{*-1};
- Breakdown and near-breakdown issues;
- Computational approach: discretization, loss of orthogonality, stability, convergence, inexact approaches, error estimates ....

Future developments and open problems

- Convergence properties of the approximation;
- Explicit formulation of the $*$-inverses β_n^{*-1};
- Breakdown and near-breakdown issues;
- Computational approach: discretization, loss of orthogonality, stability, convergence, inexact approaches, error estimates . . . .

Thank you for your attention!