Solution Formulas for Differential Sylvester and Lyapunov Equations

Maximilian Behr Peter Benner Jan Heiland

November 21, 2018

Abstract

The differential Sylvester equation and its symmetric version, the differential Lyapunov equation, appear in different fields of applied mathematics like control theory, system theory, and model order reduction. The few available straight-forward numerical approaches if applied to large-scale systems come with prohibitively large storage requirements. This shortage motivates us to summarize and explore existing solution formulas for these equations. We develop a unifying approach based on the spectral theorem for normal operators like the Sylvester operator $\mathcal{S}(X) = AX + XB$ and derive a formula for its norm using an induced operator norm based on the spectrum of A and B. In view of numerical approximations, we propose an algorithm that identifies a suitable Krylov subspace using Taylor series and use a projection to approximate the solution. Numerical results for large-scale differential Lyapunov equations are presented in the last sections.

Contents

1. Introduction

For coefficient matrices $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{m\times m}$, an inhomogeneity $C \in \mathbb{C}^{n\times m}$, and an initial value $D \in \mathbb{C}^{n\times m}$, we consider the differential matrix equation

$$\begin{aligned} \dot{X}(t) &= AX(t) + X(t)B + C, \\ X(t_0) &= D, \end{aligned} \tag{1}$$

and provide formulas for the solution X with $X(t) \in \mathbb{C}^{n\times m}$. Equation (1) is commonly known as differential Sylvester equation (as opposed to the algebraic *Sylvester equation* $AX + BX + C = 0$). In the symmetric case that $B = A^T$, equation (1) and its algebraic counterpart is called differential (algebraic) Lyapunov equation. In what follows, we will occasionally abbreviate Sylvester or Lyapunov equations by SLE.

In particular the differential Lyapunov equation is a useful tool for stability analysis and controller design for linear time-varying systems [2]. Equilibrium points of the differential Lyapunov equation, namely solutions of the algebraic Lyapunov equation, are used to construct quadratic Lyapunov functions for asymptotically stable linear time-invariant systems [33, Thm. 7.4.7]. The controllability and observability problem for linear time-varying systems is strongly connected to the solution of the differential Lyapunov equation [9, Ch. 13-14], [19, Ch. 3-4]. Other important applications lie in model order reduction [3] or in optimal control of linear time-invariant systems on finite time horizons [30]. Despite its importance, there have been but a few efforts to solve the differential Sylvester / Lyapunov or Riccati equation numerically, see [6,7,16,21,25,26,31,36]. These algorithms are usually based on applying a time discretization and solving the resulting algebraic equations. Thus, even if the algebraic SLE are solved efficiently, the storage needed for the discrete solution at the time instances makes these direct approaches infeasible for large scale settings. Recently, Krylov subspace based methods were proposed in [12–14,24].

In an attempt to overcome this shortage, we revisit known solution formulas, develop alternative solution representations, and discuss their suitability for numerical approximations. We start with deriving a spectral decomposition for the Sylvester operator $\mathcal{S}$ which allows functional calculus. We obtain formulas for the operator norm $\|\mathcal{S}\|$ as well as for $\mathcal{S}^{-1}$ and $e^{\mathcal{S}}$. This recovers previously known solution formulas. It will turn out that, in terms of efficiency, this solution representation is not well suited for approximation in general but, in special cases, allows for the construction or computation of exact solutions.

As a step towards efficient solution approximation, we use Taylor series expansions to identify suitable Krylov subspaces. For the differential Lyapunov equation with stable coefficient matrices and symmetric low-rank factored inhomogeneity, it is well-known that the solution of the algebraic Lyapunov equation spans a Krylov subspace under these assumptions. We split the solution of the differential Lyapunov equation in a constant and time dependent part, where the constant part is the solution of an algebraic Lyapunov equation. We approximate the time dependent part using the subspace spanned by the solution of the algebraic Lyapunov equation. The resulting algorithm overcomes the essential problems with storage consumption. Numerical results are presented in the Section 7 and Appendices A and B.

2. Preliminaries

In this section, we introduce the considered equations and the Sylvester operator, set the notation, and recall basic results.

The spectrum of a matrix $A \in \mathbb{C}^{n\times n}$ is denoted by $\Lambda(A)$. Generally, the spectrum is a subset of $\mathbb{C}$. A matrix is called stable if its spectrum is contained in the left open half plane $\mathbb{C}^-$. The Frobenius inner product on $\mathbb{C}^{n\times m}$ is given by $\langle A, B\rangle_F := \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m} A_{i,j}\overline{B_{i,j}}$. The Hadamard product is $A \odot B = (A_{i,j} \cdot B_{i,j})_{\substack{i=1,\dots,n\\ j=1,\dots,m}} \in \mathbb{C}^{n\times m}$ for $A, B \in \mathbb{C}^{n\times m}$. The Hermitian transpose, transpose, conjugate are denoted by A^H, $\overline{A}$, A^T, respectively. Further, we will refer to the Kronecker delta: $\delta_{ij} = \begin{cases} 1 & \text{if } i = j,\\ 0 & \text{if } i \neq j\end{cases}$ and the Kronecker product: $A \otimes B = (A_{i,j} \cdot B)_{\substack{i=1,\dots,n\\ j=1,\dots,m}}$. The identity matrix in $\mathbb{C}^{d\times d}$ is denoted by $E_{d,d}$.

For further reference we start with recalling a general well known result on the solution which applies to the more general case of the SLE (where the coefficient matrices may depend on time):

Theorem 2.1 (Existence and Uniqueness of Solutions, [1, Thm. 1.1.1., Thm. 1.1.3., Thm. 1.1.5])**.** *Let $I \subseteq \mathbb{R}$ an open interval with $t_0 \in I$, $A \in \mathcal{C}(I, \mathbb{C}^{n\times n}), B \in \mathcal{C}(I, \mathbb{C}^{m\times m}), C \in \mathcal{C}(I, \mathbb{C}^{n\times m})$ and $D \in \mathbb{C}^{n\times m}$. The differential Sylvester equation*

$$\begin{aligned}\dot{X}(t) &= A(t)X(t) + X(t)B(t) + C(t),\\ X(t_0) &= D,\end{aligned}$$

has the unique solution $X(t) = \Phi_A(t, t_0)D\Phi_{B^H}(t, t_0)^H + \int\limits_{t_0}^{t} \Phi_A(t, s)C(s)\Phi_{B^H}(t, s)^H \mathrm{d}s$.
$\Phi_A(t, t_0)$ and $\Phi_{B^H}(t, t_0)$ are the unique state-transition matrices with respect to $t_0 \in I$ defined by

$$\begin{aligned}\dot{\Phi}_A(t, t_0) &:= \frac{\partial}{\partial t}\Phi_A(t, t_0) = A(t)\Phi_A(t, t_0),\\ \Phi_A(t_0, t_0) &\ = E_{n,n}.\\ \dot{\Phi}_{B^H}(t, t_0) &:= \frac{\partial}{\partial t}\Phi_{B^H}(t, t_0) = B(t)^H\Phi_{B^H}(t, t_0),\\ \Phi_{B^H}(t_0, t_0) &\ = E_{m,m}.\end{aligned}$$

The specification to the autonomous case with constant coefficients is straight forward by simply replacing the state transition matrices with the matrix exponentials.

Theorem 2.2.
Let $I \subseteq \mathbb{R}$ an open interval with $t_0 \in I$, $A \in \mathbb{C}^{n\times n}, B \in \mathbb{C}^{m\times m}, C \in \mathcal{C}(I, \mathbb{C}^{n\times m})$ and $D \in \mathbb{C}^{m\times n}$. The differential Sylvester equation

$$\begin{aligned}\dot{X}(t) &= AX(t) + X(t)B + C(t),\\ X(t_0) &= D,\end{aligned}$$

has the unique solution

$$X(t) = e^{A(t-t_0)}De^{B(t-t_0)} + \int\limits_{t_0}^{t} e^{A(t-s)}C(s)e^{B(t-s)}\mathrm{d}s. \tag{2}$$

Next, we state basic properties of the Sylvester operator, which, for given $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{m\times m}$, is defined through its action on an $X \in \mathbb{C}^{n\times m}$:

$$\mathcal{S}(X) = AX + XB. \tag{3}$$

The Sylvester operator $\mathcal{S}$ has been thoroughly studied in [10,22,23,35]. Among others, it has been shown that the eigenvalues and eigenvectors of the Sylvester operator $\mathcal{S}$ can be expressed in terms of the eigenvalues and eigenvectors of A and B, cf. [1, Rem. 1.1.2.], [3, Ch. 6.1.1], [20].
In view of rewriting the solution formula (2), we state the following lemma:

Lemma 2.1 (Sylvester Operator $\mathcal{S}$).
For the Sylvester operator $\mathcal{S}\colon \mathbb{C}^{n\times m} \to \mathbb{C}^{n\times m}$ and its partial realizations $\mathcal{H}, \mathcal{V}\colon \mathbb{C}^{n\times m} \to \mathbb{C}^{n\times m}$, $\mathcal{H}(X) = AX$ and $\mathcal{V}(X) = XB$, it holds that:

- *$\mathcal{S} = \mathcal{H} + \mathcal{V}$ and $\mathcal{H}\mathcal{V} = \mathcal{V}\mathcal{H}$,*
- *$e^{t\mathcal{S}} = e^{t\mathcal{H}}e^{t\mathcal{V}}$ for all $t \in \mathbb{R}$,*

for any $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{m\times m}$.

Proof. The first claim can be confirmed by direct computations. The second claim is a standard result for commuting linear operators. □

By Lemma 2.1, formula (2) rewrites as

$$\begin{aligned} X(t) &= e^{tA}De^{tB} + \int_{t_0}^{t} e^{(t-s)A}C(s)e^{(t-s)B}\mathrm{d}s = e^{(t-t_0)\mathcal{H}}e^{(t-t_0)\mathcal{V}}(D) + \int_{t_0}^{t} e^{(t-s)\mathcal{H}}e^{(t-s)\mathcal{V}}(C(s))\mathrm{d}s \\ &= e^{(t-t_0)\mathcal{S}}(D) + \int_{t_0}^{t} e^{(t-s)\mathcal{S}}(C(s))\mathrm{d}s. \end{aligned} \tag{4}$$

3. Spectral Decomposition of the Sylvester Operator

In this section we show that the Sylvester operator $\mathcal{S}$, as defined in (3), is a normal operator if A and B are diagonalizable and if a suitably chosen inner product on a Hilbert space is considered. The inner product depends on the decomposition of A and B. Nevertheless, this approach will enable us to apply the spectral theorem and to derive a solution formula for the differential and algebraic SLE. This resembles the formulas of [11, Ch. 4.1.1], [18], [34]. Those results were obtained by inserting the spectral decomposition into the SLE and by applying suitable algebraic manipulations and using the unrolled Kronecker representation of the SLE. Our strategy is to decompose the operator $\mathcal{S}$ first and then using functional calculus to obtain formulas for $e^{\mathcal{S}}$ and $\mathcal{S}^{-1}$. The eigenspaces of $\mathcal{S}$ can be constructed from the eigenspaces of A and B. The choice of the inner product ensures that the eigenvectors are orthonormal and $\mathcal{S}$ becomes a normal operator.

Lemma 3.1 (Inner product for $\mathcal{S}$).
Let $A \in \mathbb{C}^{n\times n}, B \in \mathbb{C}^{m\times m}$ be diagonalizable and $C, D \in \mathbb{C}^{n\times m}$. Furthermore let $A = UD_AU^{-1}$ and $B^H = VD_{B^H}V^{-1}$ be the spectral decompositions of A and B^H with $U \in \mathbb{C}^{n\times n}, V \in \mathbb{C}^{m\times m}$, D_A and D_{B^H} diagonal matrices containing the eigenvalues $\alpha_1, \ldots, \alpha_n$ and $\overline{\beta_1}, \ldots, \overline{\beta_m}$ of A and B^H. It holds:

(i) $\langle X, Y\rangle_{U,V} := \langle U^{-1}XV^{-H}, U^{-1}YV^{-H}\rangle_F$ is an inner product on $\mathbb{C}^{n\times m}$.

(ii) $(u_i{v_j}^H)_{\substack{i=1,\ldots,n\\ j=1,\ldots,m}}$ is an orthonormal basis of $\mathbb{C}^{n\times m}$ with respect to $\langle\cdot,\cdot\rangle_{U,V}$.

(iii) *The adjoint operator* $\mathcal{S}^*: \mathbb{C}^{n\times m} \to \mathbb{C}^{n\times m}$ *with respect to* $\langle\cdot,\cdot\rangle_{U,V}$ *is* $\mathcal{S}^*(X) = U\overline{D_A}U^{-1}X + XV^{-H}D_{B^H}V^H$.

(iv) $\mathcal{S}$ *is a normal operator with respect to* $\langle\cdot,\cdot\rangle_{U,V}$.

Proof. Note that $\langle\cdot,\cdot\rangle_{U,V}$ defines an inner product since

$$\langle u_i v_j^H, u_k v_l^H\rangle_{U,V} = \langle U^{-1}u_i v_j^H V^{-H}, U^{-1}u_k v_l^H V^{-H}\rangle_F = \langle e_i e_j^H, e_k e_l^H\rangle_F = \delta_{i,k}\delta_{j,l}.$$

The matrices $u_i v_j^H \in \mathbb{C}^{n\times m}$ are orthogonal with respect to $\langle\cdot,\cdot\rangle_{U,V}$ and therefore linearly independent. Because $\dim(\mathbb{C}^{n\times m}) = n\cdot m$, the tuple $(u_i v_j^H)_{\substack{i=1,\dots,n\\ j=1,\dots,m}}$ is an orthonormal basis of $\mathbb{C}^{n\times m}$.
The following two computations show that $\mathcal{S}$ commutes with its adjoint $\mathcal{S}^*$.
Let $X, Y \in \mathbb{C}^{n\times m}$.

$$\begin{aligned}
\langle S(X), Y\rangle_{U,V} &= \langle AX + XB, Y\rangle_{U,V} = \langle AX, Y\rangle_{U,V} + \langle XB, Y\rangle_{U,V}\\
&= \langle U^{-1}AXV^{-H}, U^{-1}YV^{-H}\rangle_F + \langle U^{-1}XBV^{-H}, U^{-1}YV^{-H}\rangle_F\\
&= \langle D_A U^{-1}XV^{-H}, U^{-1}YV^{-H}\rangle_F + \langle U^{-1}XV^{-H}\overline{D_{B^H}}, U^{-1}YV^{-H}\rangle_F\\
&= \langle U^{-1}XV^{-H}, \overline{D_A}U^{-1}YV^{-H} + U^{-1}YV^{-H}D_{B^H}\rangle_F\\
&= \langle U^{-1}XV^{-H}, U^{-1}\left(U\overline{D_A}U^{-1}Y + YV^{-H}D_{B^H}V^H\right)V^{-H}\rangle_F\\
&= \langle X, \left(U\overline{D_A}U^{-1}Y + YV^{-H}D_{B^H}V^H\right)\rangle_{U,V}\\
&= \langle X, \mathcal{S}^*(Y)\rangle_{U,V}.
\end{aligned}$$

Therefore, the adjoint of $\mathcal{S}$ is $\mathcal{S}^*(X) = U\overline{D_A}U^{-1}X + XV^{-H}D_{B^H}V^H$. Moreover,

$$\begin{aligned}
\mathcal{S}\mathcal{S}^*(X) &= \mathcal{S}(U\overline{D_A}U^{-1}X + XV^{-H}D_{B^H}V^H) = \mathcal{S}(U\overline{D_A}U^{-1}X) + \mathcal{S}(XV^{-H}D_{B^H}V^H)\\
&= UD_A\overline{D_A}U^{-1}X + U\overline{D_A}U^{-1}XV^{-H}\overline{D_{B^H}}V^H\\
&\quad + UD_AU^{-1}XV^{-H}D_{B^H}V^H + XV^{-H}D_{B^H}\overline{D_{B^H}}V^H\\
&= U\overline{D_A}D_AU^{-1}X + U\overline{D_A}U^{-1}XV^{-H}\overline{D_{B^H}}V^H\\
&\quad + UD_AU^{-1}XV^{-H}D_{B^H}V^H + XV^{-H}\overline{D_{B^H}}D_{B^H}V^H\\
&= \mathcal{S}^*(UD_AU^{-1}X) + \mathcal{S}^*(XV^{-H}\overline{D_{B^H}}V^H) = \mathcal{S}^*\mathcal{S}(X).
\end{aligned}$$

This means $\mathcal{S}$ and $\mathcal{S}^*$ commute and, therefore, by definition, $\mathcal{S}$ is normal. □

Now that we have an inner product on a Hilbert space for which $\mathcal{S}$ is normal, the second step is to compute the spectral decomposition of $\mathcal{S}$. The spectral decomposition allows functional calculus and we get a formula for $\mathcal{S}^{-1}$ and $e^{t\mathcal{S}}$. Since for normal operators, the operator norm is its spectral radius, we directly get a formula for the induced operator norm of $\mathcal{S}$. We mention that in the case that A and B are unitarily diagonalizable, there is no need to exchange the inner product as one can take the Frobenius inner product $\langle\cdot,\cdot\rangle_F$.

Lemma 3.2 (Spectral Decomposition of $\mathcal{S}$).
Let the assumptions of Lemma 3.1 *hold. Then it holds:*

(i) $\mathcal{S}(X) = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}(\alpha_i + \beta_j)\langle X, u_i v_j^H\rangle_{U,V} u_i v_j^H = U\left((\alpha_i + \beta_j)_{\substack{i=1,\dots,n\\ j=1,\dots,m}} \odot U^{-1}XV^{-H}\right)V^H.$

(ii) $\|\mathcal{S}\| = \max\limits_{X\in\mathbb{C}^{n\times m}\setminus\{0\}} \frac{\|\mathcal{S}(X)\|_{U,V}}{\|X\|_{U,V}} = \max\limits_{i,j}|\alpha_i + \beta_j|$, *where* $\|X\|_{U,V} = \sqrt{\langle X, X\rangle_{U,V}}$.

(iii) $\mathcal{S}^{-1}(X) = U\left(\left(\frac{1}{\alpha_i+\beta_j}\right)_{\substack{i=1,\dots,n\\ j=1,\dots,m}} \odot U^{-1}XV^{-H}\right)V^H$ *and*

$$e^{t\mathcal{S}}(X) = U\left(\left(e^{t(\alpha_i+\beta_j)}\right)_{\substack{i=1,\dots,n\\ j=1,\dots,m}} \odot U^{-1}XV^{-H}\right)V^H.$$

Proof.

(i) From $AU = UD_A$ and $B^HV = VD_{B^H}$, we deduce $\mathcal{S}(u_kv_l^H) = Au_kv_l^H + u_kv_l^HB = (\alpha_k+\beta_l)u_kv_l^H$. Representing $\mathcal{S}(X) \in \mathbb{C}^{n\times m}$ as well as $X \in \mathbb{C}^{n\times m}$ as a Fourier series and exploiting linearity of $\mathcal{S}$ and $\langle\cdot,\cdot\rangle_{U,V}$ yields

$$\begin{aligned}
\mathcal{S}(X) &= \sum_{i=1}^{n}\sum_{j=1}^{m}\langle \mathcal{S}(X), u_iv_j^H\rangle_{U,V}u_iv_j^H = \sum_{i=1}^{n}\sum_{j=1}^{m}\langle \mathcal{S}(\sum_{k=1}^{n}\sum_{l=1}^{m}\langle X, u_kv_l^H\rangle_{U,V}u_kv_l^H), u_iu_j^H\rangle_{U,V}u_iv_j^H\\
&= \sum_{i,k=1}^{n}\sum_{j,l=1}^{m}\langle X, u_kv_l^H\rangle_{U,V}\langle \mathcal{S}(u_kv_l^H), u_iv_j^H\rangle_{U,V}u_iv_j^H\\
&= \sum_{i,k=1}^{n}\sum_{j,l=1}^{m}(\alpha_k+\beta_l)\langle X, u_kv_l^H\rangle_{U,V}\langle u_kv_l^H, u_iv_j^H\rangle_{U,V}u_iv_j^H\\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}(\alpha_i+\beta_j)\langle X, u_iv_j^H\rangle_{U,V}u_iv_j^H = U\Big((\alpha_i+\beta_j)\langle X, u_iv_j^H\rangle_{U,V}\Big)_{\substack{i=1,\dots,n\\ j=1,\dots,m}}V^H\\
&= U\left((\alpha_i+\beta_j)_{\substack{i=1,\dots,n\\ j=1,\dots,m}} \odot \Big(\langle U^{-1}XV^{-H}, e_ie_j^H\rangle_F\Big)_{\substack{i=1,\dots,n\\ j=1,\dots,m}}\right)V^H\\
&= U\left((\alpha_i+\beta_j)_{\substack{i=1,\dots,n\\ j=1,\dots,m}} \odot U^{-1}XV^{-H}\right)V^H.
\end{aligned}$$

(ii) The claim about the norm follows from a direct application of the fundamental functional analytical result on compact normal operators, see, e.g., [37, Thm. VI.3.2].

(iii) With the spectral decomposition of $\mathcal{S}$ one can resort to functional calculus, cf. [32, Cor. 9.3.38], [37, Kor. IX.3.8], and obtain the formula for $\mathcal{S}^{-1}$ under the additional assumption that $\alpha_i + \beta_j \neq 0$.

□

Using the spectral decomposition and functional calculus we find that, under the assumptions of Lemma 3.1, the solution of the differential Sylvester equation

$$\begin{aligned}
\dot{X}(t) &= AX(t) + X(t)B + C = \mathcal{S}(X(t)) + C,\\
X(0) &= D,
\end{aligned}$$

has the form

$$\begin{aligned}
X(t) &= e^{t\mathcal{S}}(D) + \int_0^t e^{(t-s)\mathcal{S}}(C)\,\mathrm{d}s\\
&= U\left(\left(e^{t(\alpha_i+\beta_j)}\right)_{\substack{i=1,\dots,n\\ j=1,\dots,m}} \odot U^{-1}DV^{-H}\right)V^H + \int_0^t U\left(\left(e^{(t-s)(\alpha_i+\beta_j)}\right)_{\substack{i=1,\dots,n\\ j=1,\dots,m}} \odot U^{-1}CV^{-H}\right)V^H\mathrm{d}s
\end{aligned}$$

$$= U \left(\left(e^{t(\alpha_i+\beta_j)} \right)_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} \odot U^{-1} D V^{-H} + \left(\int\limits_0^t e^{(t-s)(\alpha_i+\beta_j)} \mathrm{d}s \right)_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} \odot U^{-1} C V^{-H} \right) V^H, \tag{5}$$

with the involved scalar integrals given explicitly as:

$$\int\limits_0^t e^{(t-s)(\alpha_i+\beta_j)} \mathrm{d}s = \begin{cases} \frac{e^{t(\alpha_i+\beta_j)}-1}{\alpha_i+\beta_j} & \text{if } \alpha_i+\beta_j \neq 0, \\ t & \text{if } \alpha_i+\beta_j = 0 \end{cases}.$$

4. Variation of Constants

The application of the variation of constants formula leads to yet another solution formula for the SLE (1).

Lemma 4.1 (Variations of Constants, [9, Ch. 13]).
Let $A \in \mathbb{C}^{n\times n}, B \in \mathbb{C}^{m\times m}, C \in \mathbb{C}^{m\times n}, D \in \mathbb{C}^{m\times n}$ with $\Lambda(A) \cap \Lambda(-B) = \emptyset$. The differential Sylvester equation

$$\begin{aligned} \dot{X}(t) &= AX(t) + X(t)B + C = \mathcal{S}(X(t)) + C, \\ X(0) &= D, \end{aligned}$$

has the solution

$$X(t) = e^{t\mathcal{S}}(D) + \mathcal{S}^{-1}(-C) - e^{t\mathcal{S}}\mathcal{S}^{-1}(-C). \tag{6}$$

Proof. Because of $\Lambda(A) \cap \Lambda(-B) = \emptyset$, the inverse $\mathcal{S}^{-1}$ exists and we can rewrite the solution formula (4) as

$$\begin{aligned} X(t) &= e^{t\mathcal{S}}(D) + \int\limits_0^t e^{(t-s)\mathcal{S}}(C)\mathrm{d}s = e^{t\mathcal{S}}(D) + \left[-\mathcal{S}^{-1} e^{(t-s)\mathcal{S}}(C) \right]_{s=0}^{s=t} \\ &= e^{t\mathcal{S}}(D) + \mathcal{S}^{-1}(-C) - \mathcal{S}^{-1} e^{t\mathcal{S}}(-C) \end{aligned}$$

and confirm that $X(0) = D + \mathcal{S}^{-1}(-C) - \mathcal{S}^{-1}(-C) = D$. □

From formula (6), we find that the solution can be written as the solution of the algebraic Sylvester equation and a time dependent part. We will make use of this fact in the numerical scheme that we propose in Section 6.

5. Solution as Taylor Series

In this section we use Taylor series to derive a solution formula. From this we can read off suitable Krylov subspaces for our projection approach in the next section.

Lemma 5.1 (Taylor Series Solution Formula).
Let $A \in \mathbb{C}^{n\times n}, B \in \mathbb{C}^{m\times m}, C \in \mathbb{C}^{m\times n}, D \in \mathbb{C}^{m\times n}$. The differential Sylvester equation

$$\begin{aligned} \dot{X}(t) &= AX(t) + X(t)B + C = \mathcal{S}(X(t)) + C, \\ X(0) &= D, \end{aligned}$$

has the unique solution

$$X(t) = D + \sum_{k=1}^{\infty} \frac{t^k}{k!}(\mathcal{S}^k(D) + \mathcal{S}^{k-1}(C)). \tag{7}$$

Proof.

$$\begin{aligned}
||D|| + \sum_{k=1}^{\infty} ||\frac{t^k}{k!}(\mathcal{S}^k(D) + \mathcal{S}^{k-1}(C))|| &\leq ||D|| + \sum_{k=1}^{\infty} \frac{|t|^k}{k!}(||\mathcal{S}^k(D)|| + ||\mathcal{S}^{k-1}(C)||) \\
&\leq ||D|| + \sum_{k=1}^{\infty} \frac{|t|^k}{k!}(||\mathcal{S}||^k ||D|| + ||\mathcal{S}||^{k-1}||C||) \\
&= ||D|| e^{|t|||\mathcal{S}||} + ||C|| \sum_{k=0}^{\infty} \frac{|t|^{k+1}}{(k+1)!} ||\mathcal{S}||^k \\
&\leq ||D|| e^{|t|||\mathcal{S}||} + |t|||C|| e^{|t|||\mathcal{S}||}.
\end{aligned}$$

The series converges absolutely and since $(\mathbb{C}^{n\times m}, ||\cdot||)$ is a Banach space, the series converges for every $t \in \mathbb{R}$. Therefore its radius of convergence is infinity, X is continuously differentiable and can be differentiated term-wise. Since, furthermore,

$$X(0) = D$$

and

$$\begin{aligned}
\dot{X}(t) &= \sum_{k=1}^{\infty} \frac{t^{k-1}}{(k-1)!}(\mathcal{S}^k(D) + \mathcal{S}^{k-1}(C)) \\
&= \sum_{k=0}^{\infty} \frac{t^k}{k!}(\mathcal{S}^{k+1}(D) + \mathcal{S}^k(C)) = \mathcal{S}(D) + \sum_{k=1}^{\infty} \frac{t^k}{k!}(\mathcal{S}^{k+1}(D) + \mathcal{S}^k(C)) + C \\
&= \mathcal{S}(D + \sum_{k=1}^{\infty} \frac{t^k}{k!}(\mathcal{S}^k(D) + \mathcal{S}^{k-1}(C))) + C = \mathcal{S}(X(t)) + C,
\end{aligned}$$

$X(t)$ is the unique solution. □

If we assume that D and C are given in factored form, then we can exploit this to rewrite the truncated series in a closed form of a matrix product.

Remark 5.1.
Let $D = D_1 D_2^H$ *and* $C = C_1 C_2^H$ *with* $D_1 \in \mathbb{C}^{n\times d}, D_2 \in \mathbb{C}^{m\times d}, C_1 \in \mathbb{C}^{n\times c}$ *and* $C_2 \in \mathbb{C}^{m\times c}$. *Then, having truncated the two parts of the series* (7) *after* m_1 *and* m_2 *summands, respectively, we can rewrite the solution approximation as*

$$\begin{aligned}
X_{m_1,m_2}(t) &= \sum_{k=0}^{m_1} \frac{t^k}{k!}\mathcal{S}^k(D) + \sum_{k=1}^{m_2} \frac{t^k}{k!}\mathcal{S}^{k-1}(C) \\
&= \sum_{k=0}^{m_1} \frac{t^k}{k!}\mathcal{S}^k(D_1 D_2^H) + \sum_{k=1}^{m_2} \frac{t^k}{k!}\mathcal{S}^{k-1}(C_1 C_2^H) \\
&= \sum_{k=0}^{m_1} \frac{t^k}{k!}(\mathcal{H} + \mathcal{V})^k(D_1 D_2^H) + \sum_{k=1}^{m_2} \frac{t^k}{k!}(\mathcal{H} + \mathcal{V})^{k-1}(C_1 C_2^H) \\
&= \sum_{k=0}^{m_1} \sum_{i=0}^{k} \frac{t^k}{k!}\binom{k}{i}\mathcal{H}^{k-i}\mathcal{V}^i(D_1 D_2^H) + \sum_{k=1}^{m_2} \sum_{i=0}^{k-1} \frac{t^k}{k!}\binom{k-1}{i}\mathcal{H}^{k-1-i}\mathcal{V}^i(C_1 C_2^H)
\end{aligned}$$

$$= \sum_{k=0}^{m_1} \sum_{i=0}^{k} \frac{t^k}{k!} \binom{k}{i} A^{k-i} D_1 D_2^H B^i + \sum_{k=1}^{m_2} \sum_{i=0}^{k-1} \frac{t^k}{k!} \binom{k-1}{i} A^{k-1-i} C_1 C_2^H B^i,$$

with the explicit representation of the sums

$$\sum_{k=0}^{m_1} \sum_{i=0}^{k} \frac{t^k}{k!} \binom{k}{i} A^{k-i} D_1 D_2^H B^i =$$

$$\Big[D_1, AD_1, \ldots, A^{m_1} D_1\Big] \left(\begin{bmatrix} \frac{t^0}{0!}\binom{0}{0} & \frac{t^1}{1!}\binom{1}{1} & \frac{t^2}{2!}\binom{2}{2} & \cdots & \frac{t_1^m}{m_1!}\binom{m_1}{m_1} \\ \frac{t^1}{1!}\binom{1}{0} & \iddots & \iddots & \iddots & 0 \\ \frac{t^2}{2!}\binom{2}{0} & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & 0 \\ \frac{t^{m_1}}{m_1!}\binom{m_1}{0} & 0 & \cdots & 0 & 0 \end{bmatrix} \otimes E_{d,d} \right) \begin{bmatrix} D_2^H \\ D_2^H B \\ \vdots \\ D_2^H B^{m_1} \end{bmatrix}$$

and

$$\sum_{k=1}^{m_2} \sum_{i=0}^{k-1} \frac{t^k}{k!} \binom{k-1}{i} A^{k-1-i} C_1 C_2^H B^i =$$

$$\Big[C_1, AC_1, \ldots, A^{m_2-1} C_1\Big] \left(\begin{bmatrix} \frac{t^1}{1!}\binom{0}{0} & \frac{t^2}{2!}\binom{1}{1} & \frac{t^3}{3!}\binom{2}{2} & \cdots & \frac{t^{m_2}}{m_2!}\binom{m_2-1}{m_2-1} \\ \frac{t^2}{2!}\binom{1}{0} & \iddots & \iddots & \iddots & 0 \\ \frac{t^3}{3!}\binom{2}{0} & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & 0 \\ \frac{t^{m_2}}{m_2!}\binom{m_2-1}{0} & 0 & \cdots & 0 & 0 \end{bmatrix} \otimes E_{c,c} \right) \begin{bmatrix} C_2^H \\ C_2^H B \\ \vdots \\ C_2^H B^{m_2-1} \end{bmatrix}.$$

6. Feasible Numerical Solution Approaches

In this section, we briefly note that, for various reasons, all presented solution representations are not feasible for a straight-forward numerical approximation, in particular in a large-scale sparse setting.

A common reason is that none of the formulas supports a sparse representation of the solutions such that exorbitant amounts of memory will be required.

Limitations in memory will doubly affect the solution representation through the spectral decomposition (5) since also the basis matrices U and V are generically dense matrices. Apart from that, the computation of a spectral decomposition is typically computationally expensive and can be ill conditioned. Nonetheless, the spectral decomposition formula is useful to construct exact solutions for given coefficients with known spectral decompositions.

Another issue is the unfeasible computation of the full matrix exponential in all variants (2), (4), and (6) of the *variation of constants* formula. A possible remedy is the approximation the action of the matrix exponential on a low-rank matrix in a Krylov subspace.

The approach to the solution via a Taylor series (see Section 5) seems best suited for the large-scale case since, at least in the symmetric case, the formulas provided in Remark 5.1 allow for a solution representation in factored form with the original coefficients. One problem here is that the truncated Taylor series only leads to good approximations locally around the point of expansion.

We will, however, exploit and combine certain parts of the solution representations to propose an algorithm for fast and memory efficient solution approximations.

We consider the stable, linear time-invariant case, i.e. we assume that $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, and

$\Lambda(A) \subseteq \mathbb{C}^-$. We consider the differential Lyapunov equation

$$\dot{X}(t) = A^T X(t) + X(t)A + BB^T, \tag{8a}$$
$$X(0) = 0. \tag{8b}$$

By Lemma 4.1, we have that the solution splits into a constant part and a time dependent part. From Remark 5.1, we infer that the solution is contained in a Krylov subspace. We combine both observations in the following numerical solution approach:

1. Factors of the time constant Part as Krylov space basis With A stable, the associated algebraic Lyapunov $A^T X + XA + BB^T = 0$ has a unique symmetric positive semi-definite solution X_∞ that can be written in factored form $X_\infty = Z_\infty Z_\infty^T$, with $Z_\infty \in \mathbb{R}^{n\times q}$ and $\operatorname{rank}(Z_\infty) = q \leq n$. Moreover, since A is stable, it holds (see [9, Ch. 13]) that

$$\operatorname{range}(Z_\infty) = \operatorname{range}(X_\infty) = \operatorname{range}\left(\left[B, A^T B, \dots, (A^T)^{n-1} B\right]\right).$$

2. Factors of the time dependent part evolve in the same Krylov space With $X(t) = X_\infty + \tilde{X}(t)$, we obtain that

$$\begin{aligned} \dot{\tilde{X}}(t) &= A^T \tilde{X}(t) + \tilde{X}(t)A, \\ \tilde{X}(0) &= -X_\infty, \end{aligned}$$

where $\tilde{X}(t)$ is given by $\tilde{X}(t) = -e^{tA^T} X_\infty e^{tA} = -e^{tA^T} Z_\infty Z_\infty^T e^{tA} =: -\tilde{Z}(t)\tilde{Z}(t)^T$.

3. Orthogonalize the basis and compute the time dependent factors By means of the singular value decomposition of Z_∞, we obtain an orthogonal matrices Q_∞ and V_∞ with $\operatorname{range}(Q_\infty) = \operatorname{range}(Z_\infty)$, with $Z_\infty = Q_\infty S_\infty V_\infty^T$, and with $Q_\infty \in \mathbb{R}^{n\times q}$, $S_\infty \in \mathbb{R}^{q\times q}$, $V_\infty \in \mathbb{R}^{q\times q}$. Like Z_∞, the columns of Q_∞ span an A^T invariant subspace and with $Q_\infty^T Q_\infty = E_{q,q}$ it holds that

$$\begin{aligned} A^T Q_\infty &= Q_\infty Q_\infty^T A^T Q_\infty, \\ e^{tA^T} Q_\infty &= Q_\infty e^{tQ_\infty^T A^T Q_\infty}, \end{aligned}$$

from which we confer that

$$\begin{aligned} \tilde{Z}(t)\tilde{Z}(t)^T &= \left(e^{tA^T} Z_\infty\right)\left(e^{tA^T} Z_\infty\right)^T = \left(e^{tA^T} Q_\infty S_\infty\right)\left(e^{tA^T} Q_\infty S_\infty\right)^T \\ &= \left(Q_\infty e^{tQ_\infty^T A^T Q_\infty} S_\infty\right)\left(Q_\infty e^{tQ_\infty^T A^T Q_\infty} S_\infty\right)^T. \end{aligned}$$

We define $z(t) = e^{tQ_\infty^T A^T Q_\infty} S_\infty \in \mathbb{R}^{q\times q}$ and find that z can be obtained by solving

$$\dot{z}(t) = Q_\infty^T A^T Q_\infty z(t) \tag{9a}$$
$$z(0) = S_\infty, \tag{9b}$$

which is a matrix valued ODE that can be solved column-wise or by computing the matrix exponential $e^{tQ_\infty^T A^T Q_\infty}$.

The solution of the differential Lyapunov equation is, thus, given by $X(t) = Z_\infty Z_\infty^T - Q_\infty z(t) z(t)^T Q_\infty^T$.

Remark 6.1.

The differential equation for z is of size $q \times q$, which can be much smaller than n, if the solution of the algebraic Lyapunov equation $X_\infty = Z_\infty Z_\infty^T$ has low-rank. Moreover, the orthogonalization of the basis allows for the detection of the numerical rank and for a compression of Z_∞ through truncating

singular values that are smaller than a certain threshold.
We further note that, with minor adjustments, all arguments also hold for the generalized differential Lyapunov equation

$$\begin{aligned} M^T\dot{X}(t)M &= A^TX(t)M + M^TX(t)A + BB^T, \\ X(0) &= 0, \end{aligned}$$

with $M \in \mathbb{R}^{n\times n}$ nonsingular that can accommodate, e.g., a mass matrix from a finite element discretization.

In summary, the proposed approach reads as written down in Algorithm 1.

Algorithm 1 Projection approach for generalized differential Lyapunov equations

```
Input: $M, A \in \mathbb{R}^{n\times n}$ with $\Lambda(AM^{-1}) \subseteq \mathbb{C}_-$ and $B \in \mathbb{R}^{n\times p}$.
Output: $X(t) = Z_\infty Z_\infty^T - Q_\infty z(t)z(t)^T Q_\infty^T$ that approximates the solution to
    $M^T\dot{X}(t)M = A^TX(t)M + M^TX(t)A + BB^T,\ X(0) = 0$.

% Solve Lyapunov equation:
$A^TX_\infty M + M^TX_\infty A = -BB^T$ for $X_\infty \approx Z_\infty Z_\infty^T$ and $Z_\infty \in \mathbb{R}^{n\times q}$.

% Compute singular value decomposition:
$[Q_\infty, S_\infty, \sim] = \mathrm{svd}(Z_\infty, 0)$.

% Set tolerance to largest singular value times machine epsilon:
$tol = \varepsilon_{\mathrm{machine}} \cdot S_\infty(1,1)$.

% Truncate all singular values smaller than tolerance and get truncated low-rank factor:
$idx = \mathrm{diag}(S_\infty) \geq tol$.
$S_\infty \leftarrow S_\infty(idx, idx)$.
$Q_\infty \leftarrow Q_\infty(:, idx)$.
$Z_\infty \leftarrow Q_\infty S_\infty$.

% Compute projected system and solve:
if $M$ is symmetric positive definite then
    $M_F = Q_\infty^T M^T Q_\infty$.
    $A_F = Q_\infty^T A^T Q_\infty$.
else
    $M_F = E$.
    $A_F = Q_\infty^T M^{-T} A^T Q_\infty$.
end if

for k=1,..., cols ($S_\infty$) do
    Solve: $M_F\dot{z}(:,k)(t) = A_F z(:,k)(t),\ z(:,k)(0) = S_\infty(:,k)$.
end for
```

7. Numerical Results

7.1. Setup

To quantify and illustrate the performance of Algorithm 1, we consider differential Lyapunov equations that are used to define optimal controls for a finite element discretization of a heat equation; see [8] for the model description. Namely, we solve the differential Lyapunov equations:

$$M\dot{X}(t)M^T = AX(t)M^T + MX(t)A^T + BB^T, \quad X(0) = 0. \tag{DLE–1}$$

$$M^T\dot{X}(t)M = A^TX(t)M + M^TX(t)A + C^TC, \quad X(0) = 0. \tag{DLE–2}$$

that are defined through matrices M, $A \in \mathbb{R}^{n\times n}$ that are symmetric, M is positive definite and A stable, $B \in \mathbb{R}^{n\times 7}$ and $C \in \mathbb{R}^{6\times n}$. For computing the error, we precomputed the spectral decomposition of (A, M) and constructed the exact solution by means of the formula from Lemma 3.2. The memory consuming computation of the spectral decomposition was done on a compute server with 4 × Xeon® CPU E7–8837 @ 2.67GHz with 8 cores and 1 TB Ram and MATLAB® 2015b. All other computations were carried out on a machine with 2 × Xeon® CPU E5–2640 v3 @ 2.60GHz with 8 Cores and 64 GB Ram and MATLAB 2017a. We have used the low-rank ADI iteration implemented in MEX-M.E.S.S. [4] to solve the algebraic Lyapunov equations; as required for Algorithm 1 (Step 2).

We solve the resulting projected ODE system column-wise using MATLAB ODE solvers `ode45`, `ode23`, `ode113`, `ode15s`, `ode23s`, `ode23t` and `ode23tb`, with the parameters for the `odeset` function set as follows:

- `RelTol`: $1 \cdot 10^{-9}$
- `AbsTol`: $1 \cdot 10^{-10}$
- `Stats`: `off`
- `NormControl`: `off`
- `BDF`: `on`
- `Jacobian`: A_F
- `JPattern`: `logical` (A_F)
- `Mass`: M_F
- `MassSingular`: `no`
- `MStateDependence`: `none`

As the time interval, we considered $[0, 4500]$.

7.2. Projection Approach

The initial step of Algorithm 1 requires the solutions to the associated algebraic Lyapunov equations. For this task we call MEX-M.E.S.S. that iteratively computes the solutions up to the following absolute and relative residuals

$$||AZ_\infty Z_\infty^T M^T + MZ_\infty Z_\infty^T A^T + BB^T||_2 \text{ or } ||A^TZ_\infty Z_\infty^T M + M^TZ_\infty Z_\infty^T A + C^TC||_2.$$

and

$$\frac{||AZ_\infty Z_\infty^T M^T + MZ_\infty Z_\infty^T A^T + BB^T||_2}{||BB^T||_2} \text{ or } \frac{||A^TZ_\infty Z_\infty^T M + M^TZ_\infty Z_\infty^T A + C^TC||_2}{||C^TC||_2}.$$

The achieved values for the different test setups as well as the number of columns of the corresponding Z_∞ after truncation (see Step 7 of Algorithm 1), that define the dimension of the reduced model for the time dependent part, are listed in Tables 1 and 2.

size	size of $z(t)$	absolute residual	relative residual
1357	261×261	$3.413488 \cdot 10^{-25}$	$7.748357 \cdot 10^{-12}$
5177	302×302	$1.037846 \cdot 10^{-25}$	$4.728703 \cdot 10^{-12}$
20209	376×376	$6.053185 \cdot 10^{-26}$	$5.525974 \cdot 10^{-12}$

Table 1: Residuals for $AXM^T + MXA^T + BB^T = 0$.

size	size of $z(t)$	absolute residual	relative residual
1357	230×230	$1.011843 \cdot 10^{-10}$	$8.432027 \cdot 10^{-12}$
5177	259×259	$5.595100 \cdot 10^{-11}$	$4.662583 \cdot 10^{-12}$
20209	310×310	$4.382439 \cdot 10^{-11}$	$4.382439 \cdot 10^{-12}$

Table 2: Residuals for $A^T XM + M^T XA + C^T C = 0$.

We report the absolute and relative errors

$$||X(t) - X_{ref}(t)||_2 \quad \text{and} \quad \frac{||X(t) - X_{ref}(t)||_2}{||X_{ref}(t)||_2},$$

where X is the numerical solution obtained from Algorithm 1 with various ODE solvers and where the reference solution X_{ref} was obtained from spectral decomposition of (A, M).
We plot the numerical errors and $||X_{ref}(t)||_2$ on the initial short time interval $[0, 10]$, where most of the evolution is happening, and on the full time interval $[0, 4500]$ in Appendix A, Figures 4–7, 10–13, 16–19, 22–25, 28–31 and 34–37.
In view of the performance of the different ODE solvers, we can interprete the presented numbers and plots as follows: the solver `ode15s`, which is a stiff solver of variable order, performs best in time and accuracy. Due to the stiffness, the error is oscillating for the solvers `ode45`, `ode23`, `ode113`. Note that the discrete Laplacian that is encoded in the coefficient matrix A becomes stiffer with a finer space discretization, i.e. for larger n. Accordingly, the computational times for the non-stiff solvers grow with n at a higher rate than the stiff solvers; see Figure 2.
The solution of (DLE–2) itself is large in norm what makes the relative error stagnate around the prescribed tolerance and the absolute error comparatively large; see the plots in Appendix A.2. Particularly, the non-stiff solvers (and `ode15s`) achieve this error level and the oscillations due to stiffness are dominated by the approximation error. This might be the reason, why for (DLE–2), despite the fact that the coefficient matrices are still stiff, the non-stiff solvers perform better than the stiff solvers (except `ode15s`). Nonetheless, again the computation times for the non-stiff solver grow at a higher rate with the increasing stiffness that comes with increasing n; see Figure 2.
Because $X(t) \to 0$ for $t \searrow 0$, the plots for the relative error spread out for small times.
Except `ode15s`, there is no general rule which solver performs better in terms of computational time; see Figure 2. The timings may change, when different relative and absolute error tolerances are used in the MATLAB `odeset` function.
Finally, we want to make the following remark: instead of integrating the projected ODE (9) which is linear with constant coefficients of moderate dimension, one may consider using the *Schur-Parlett* [17, Ch. 10] algorithm to compute the matrix exponential. The initialization efforts of the *Schur-Parlett* algorithm will pay off, if the matrix exponential has to be evaluated for many different values of t. Also, asymptotically, the storage requirements for $z(t)$ will be lifted, since the matrix exponential for a given t can be computed on demand. Nonetheless, we used the ODE approach, which we think is more efficient because of the sophisticated MATLAB ODE solver implementations that come, e.g, with step size selection methods integrated.
The code of the implementation with the precomputed spectral decompositions needed to construct

the exact solution is available as mentioned in Figure 1.

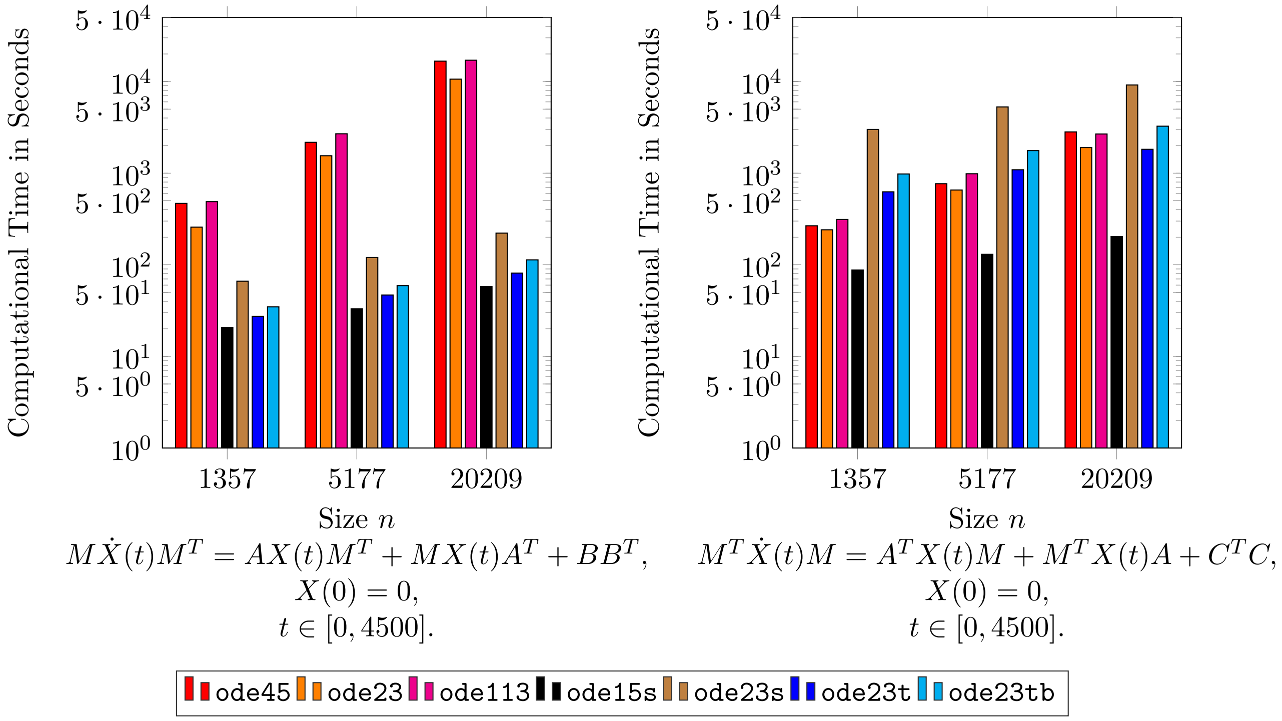
Code and Data Availability
The source code of the implementations used to compute the presented results is available from:

`doi:10.5281/zenodo.1484327`
`https://gitlab.mpi-magdeburg.mpg.de/behr/diff_lyap_eqn_solution_formulas_release`

under the GPLv2+ license and is authored by Maximilian Behr.

Figure 1: Link to code and data.

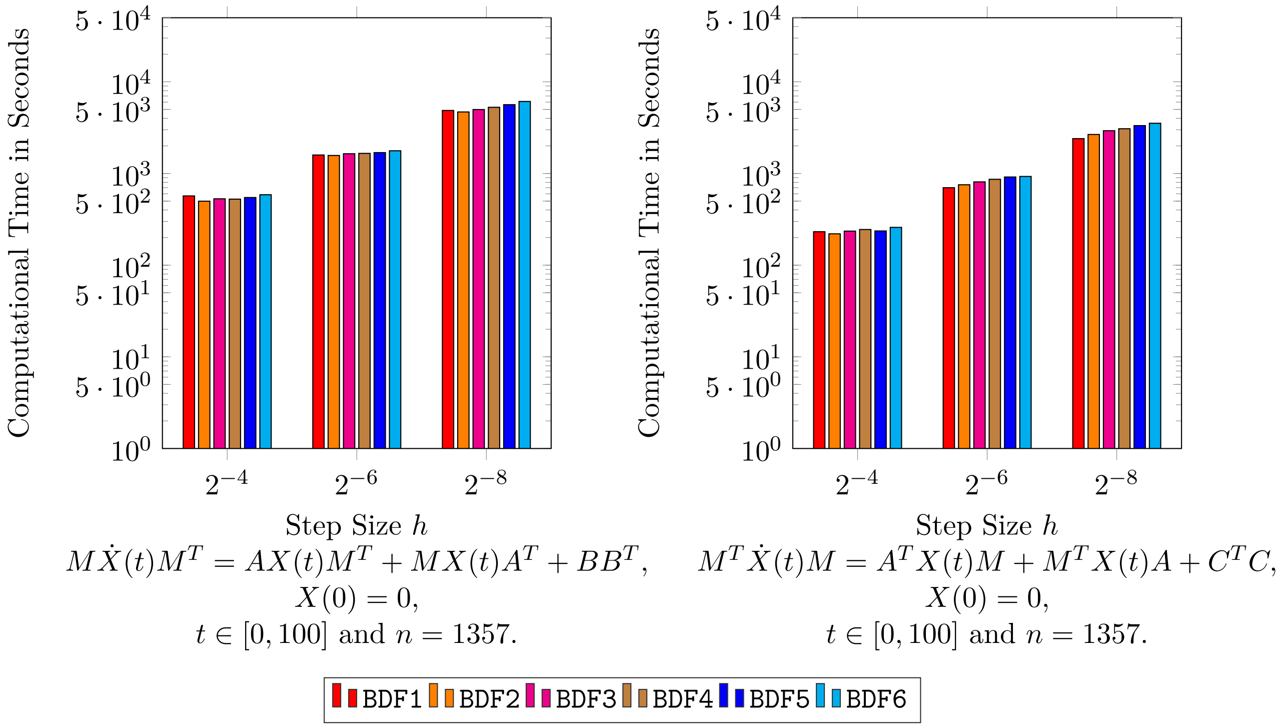
7.3. Computational Time



$M\dot{X}(t)M^T = AX(t)M^T + MX(t)A^T + BB^T,$
$X(0) = 0,$
$t \in [0, 4500].$

$M^T\dot{X}(t)M = A^TX(t)M + M^TX(t)A + C^TC,$
$X(0) = 0,$
$t \in [0, 4500].$

Figure 2: Timings of the different MATLAB ODE solvers for the solution of the projected time-dependent factors as defined in Equation (9).



$M\dot{X}(t)M^T = AX(t)M^T + MX(t)A^T + BB^T,$
$X(0) = 0,$
$t \in [0, 100]$ and $n = 1357.$

$M^T\dot{X}(t)M = A^TX(t)M + M^TX(t)A + C^TC,$
$X(0) = 0,$
$t \in [0, 100]$ and $n = 1357.$

Figure 3: Timings of the different BDF/ADI solvers.

7.4. Comparison with Backward Differentiation Formulas

To benchmark our method, we have run comparison tests with the MATLAB implementation of the Backward Differentiation Formulas / Alternating Direction Implicit (BDF/ADI) scheme as developed in [29] (see also the Appendix B for a short summary of the algorithm). The numerical experiments were conducted on the same machine, with the same MATLAB version and the same model as described in Section 7.1.

In contrast to the Algorithm 1 that needs to solve only a single algebraic Lyapunov equation for the initialization, the BDF/ADI approach solves a Lyapunov equation in every time step. Moreover, the numerical solution is stored in LDL^T-format, i.e., in terms of the factors of $X(t_k) \approx L_k D_k L_k^T$ and $L_k \in \mathbb{R}^{n \times l_k}$, which grow at least linearly with the size of n and the number of time steps. For this reason, we had to restrict our numerical experiments with BDF/ADI to the interval $[0, 100]$ and consider only the model of the smallest size $n = 1357$. As for the test of our Algorithm 1 in Appendix A, for computing the error, we used an exact solution X_{ref} based on the spectral decomposition of (A, M).

We have compared various BDF methods that we abbreviated as `BDF1`, `BDF2`, `BDF3`, `BDF4`, `BDF5` and `BDF6`, where the number denotes the order s of the method. We used the constant time step sizes $\tau_k := h \in \{2^{-4}, 2^{-6}, 2^{-8}\}$ for our computations. In Appendices B.1 and B.2, we plot the relative and absolute errors compared to the solution obtained by the spectral decomposition, c.f. Figures 40–45 and 46–51. . For comparison, we also plot the error of the numerical solution obtained by Algorithm 1 and the MATLAB ODE solver `ode15s`. We list the computational times for performing the BDF/ADI method based computations in Section 7.3.

As the actual solution X converges towards the solution of the algebraic Lyapunov equation, also the numerical errors show a decay towards a certain error level. Decreasing the step size lowers this level accordingly. This effect is visible only for methods of lower order ($s \leq 2$). For higher orders, the error level stagnates and the error rather shows oscillations, which are likely due to errors in the solution of the Lyapunov equations. Since all BDF methods are stiff solvers, there are no oscillations of higher frequency to observe. The error levels are comparable to the level reached by our approach with `ode15s`.

The computational timings for the BDF/ADI solvers are nearly the same for same step sizes; cf. Figure 3. Accordingly, the higher order methods clearly outperform the low-order methods.

As for the comparison to our approach, we note that the reported timings were for the longer time-interval $[0, 4500]$. However, since the transient behavior of the solution is confined to a short initial phase and since the ODE solvers have a step size control, a restriction to a shorter interval would not change the timings by much.

For the test case with (DLE–1), the BDF/ADI schemes achieved the same accuracy as our approach already with the coarsest step size; see Figures 40–45. Thus, we compare the timings in the left most columns in Figure 2 ($n = 1357$) and Figure 3 ($h = 2^{-4}$) to conclude that our approach with the best performing ODE solver is about 25 times faster.

As for the test case with (DLE–2), the BDF/ADI schemes reached the same accuracy only for the finest chosen step size; see Figures 46–51. Thus, comparing the right most column in Figure 3 ($h = 2^{-8}$) to the left column in the right plot of Figure 2 ($n = 1357$), we find that our Algorithm 1, again, is faster by a factor of 30.

To conclude the comparison, we add the following remarks. With the costs of solving Lyapunov equations, apart from the increased memory requirements, the BDF/ADI scheme will also become significantly more costly for larger system sizes. As opposed to our Algorithm 1 however, the BDF/ADI scheme also applies for the time varying case as well as for nonzero initial conditions. Moreover, as `ode15s` in fact uses the BDF formulas but with variable order s and variable time step, one may consider integrating similar error control mechanisms in the BDF/ADI approach to improve performance.

8. Conclusions

We presented several solution formulas for the differential Sylvester and Lyapunov equations. For the autonomous stable differential Lyapunov equation, we proposed a numerical algorithm that combined certain aspects derived from the solution representations. The main feature of the algorithm is the projection of the time dependent part onto a suitable subspace of lower dimension. Only this makes the numerical solution feasible in terms of memory requirements. As for the computational time, the projected system can be directly solved with optimized ODE solvers like the `ode-suite` in MATLAB. Moreover, its structure allows for column-wise computation of the factors and, thus, for straight-forward parallelization.

We illustrated the performance of the algorithm in an example that was derived from a finite-element discretization of a heat equation. The achieved accuracy is fully satisfactory. The greatest benefit, also in contrast to existing numerical schemes, is the low memory requirement.

The possible extension to the unstable or unsymmetric as well as to the non-autonomous case is not straightforward since the space in which the solution evolves has not been found to span a suitable invariant Krylov subspace and is possibly of high dimension. Moreover, the solution of $AX = C$ (which is the special case of $AX + XB = C$ with $B = 0$) is unlikely to span an A-invariant subspace at all, meaning that the span of the solution of the algebraic Sylvester equation is not suited for the differential equation. Thus, for the general case, a different ansatz is needed. The same is true for time-dependent coefficients or non-zero initial conditions. Here, a remedy might be structured approaches for the case that time dependence comes, e.g., from a low-rank update.

In the unstable case, apart from the fact that the algebraic Lyapunov equation may not have a unique solution, there can be modes that grow exponentially in time. For this case it may be worth investigating whether a projector that identifies the stable part of the underlying mathematical model can be efficiently incorporated in the proposed solution approach.

A. Numerical Results Projection Approach

A.1. Results for the Differential Lyapunov Equation

$$n = 1357 \text{ and } M\dot{X}(t)M^T = AX(t)M^T + MX(t)A^T + BB^T,\ X(0) = 0.$$

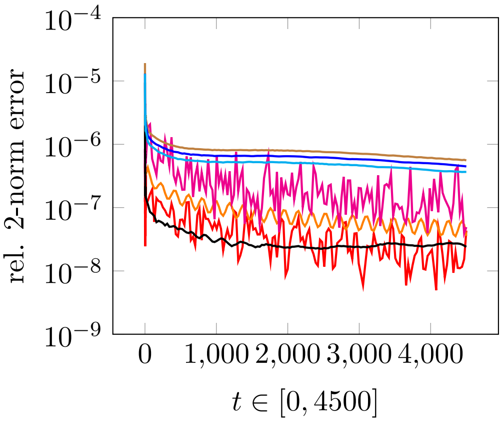


Figure 4: Relative 2-norm error of the approximation.

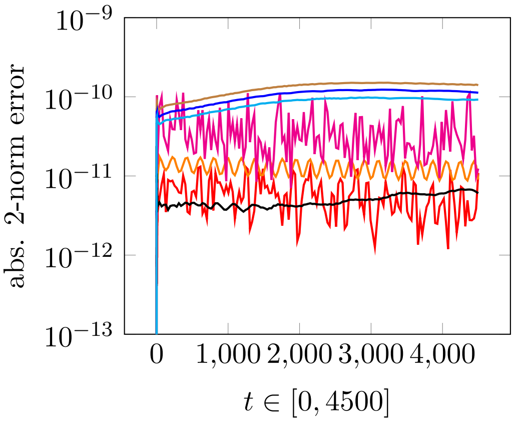


Figure 5: Absolute 2-norm error of the approximation.

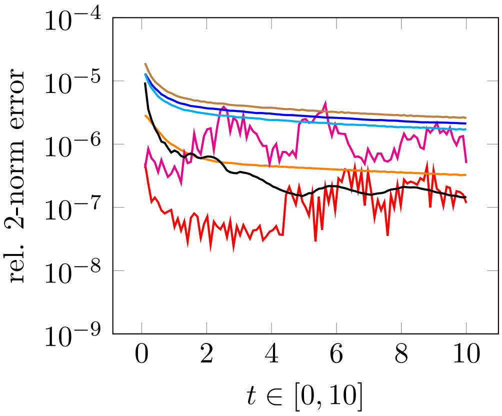


Figure 6: Relative 2-norm error of the approximation.

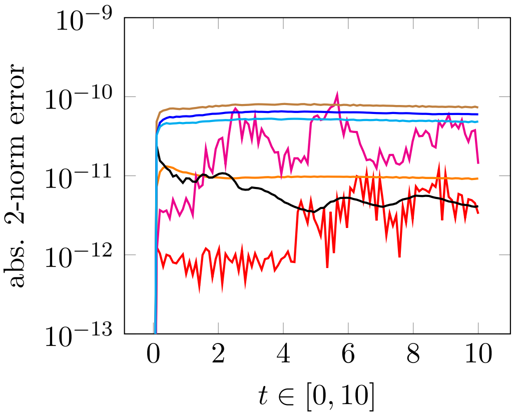


Figure 7: Absolute 2-norm error of the approximation.

— `ode45` — `ode23` — `ode113` — `ode15s` — `ode23s` — `ode23t` — `ode23tb`

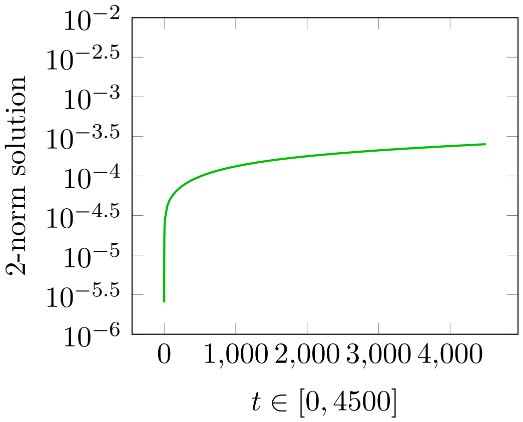


Figure 8: 2-norm of the reference solution.

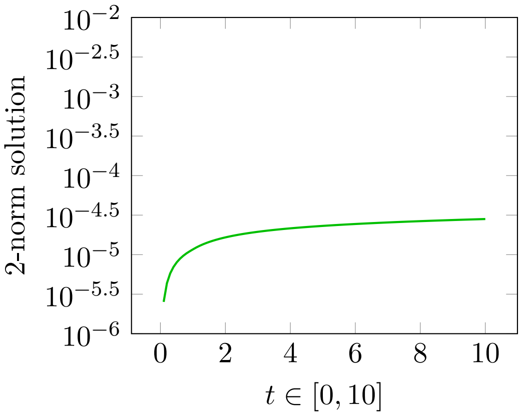


Figure 9: 2-norm of the reference solution.

$$n = 5177 \text{ and } M\dot{X}(t)M^T = AX(t)M^T + MX(t)A^T + BB^T,\ X(0) = 0.$$

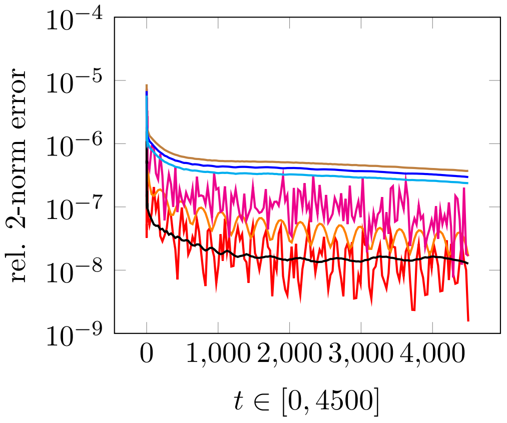


Figure 10: Relative 2-norm error of the approximation.

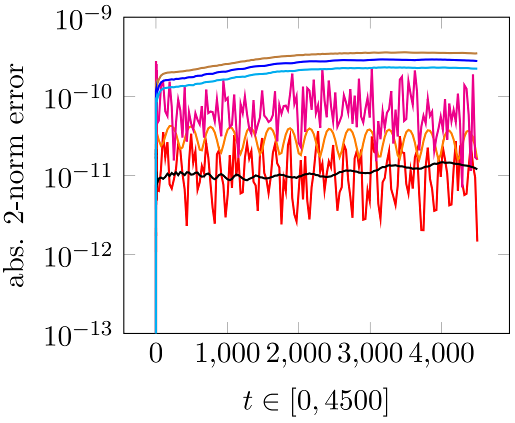


Figure 11: Absolute 2-norm error of the approximation.

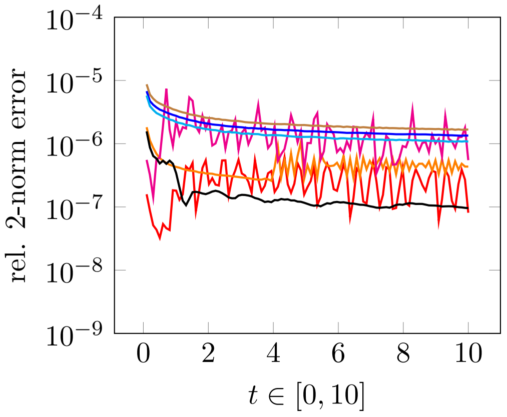


Figure 12: Relative 2-norm error of the approximation.

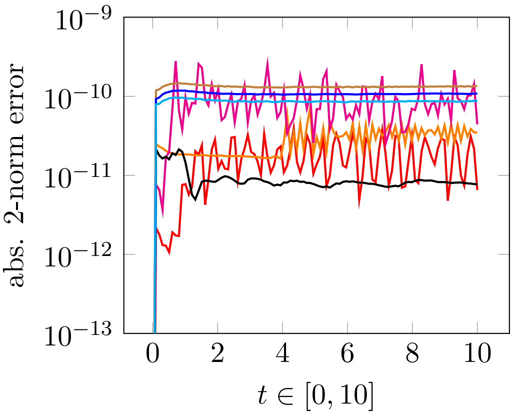
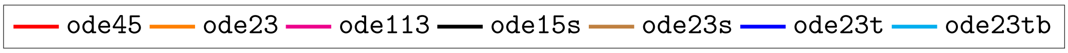


Figure 13: Absolute 2-norm error of the approximation.



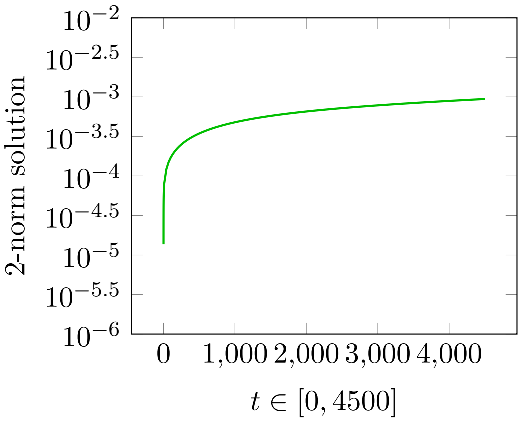


Figure 14: 2-norm of the reference solution.

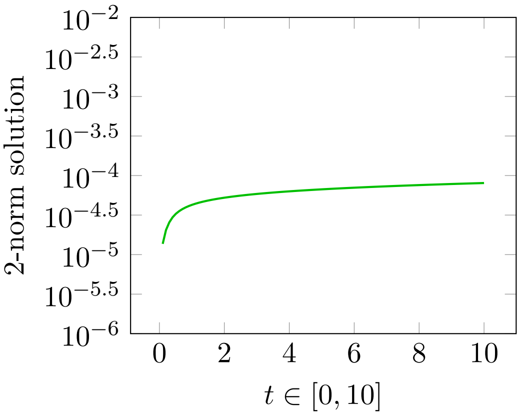


Figure 15: 2-norm of the reference solution.

$$n = 20209 \text{ and } M\dot{X}(t)M^T = AX(t)M^T + MX(t)A^T + BB^T,\ X(0) = 0.$$

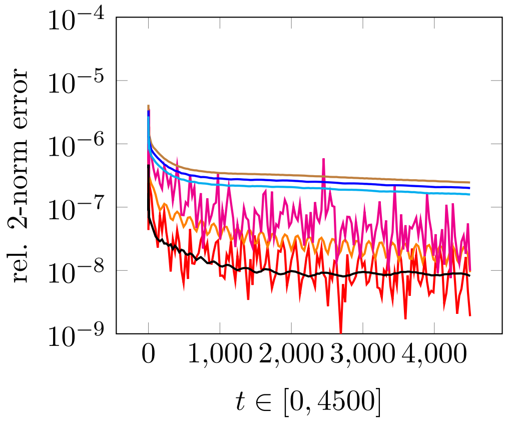


Figure 16: Relative 2-norm error of the approximation.

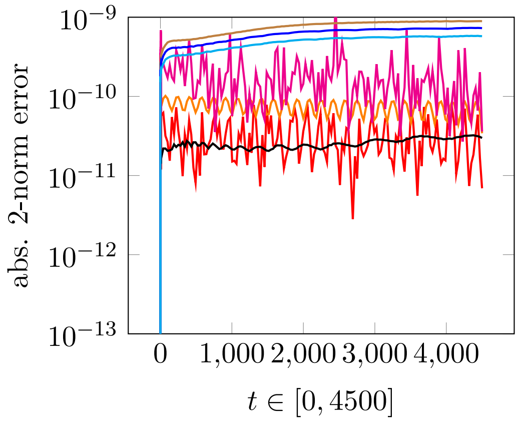


Figure 17: Absolute 2-norm error of the approximation.

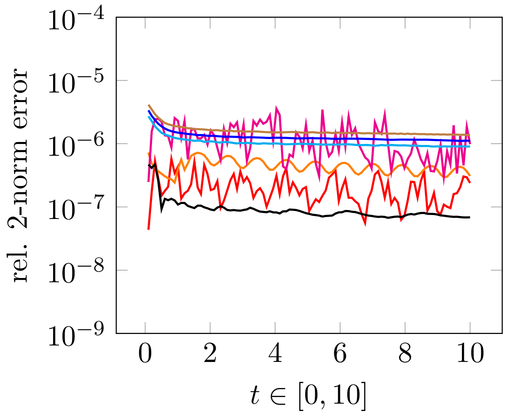


Figure 18: Relative 2-norm error of the approximation.

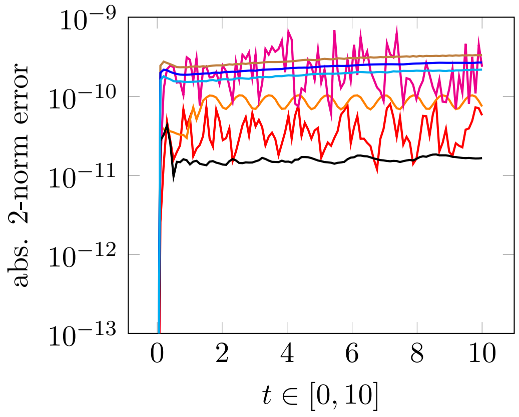


Figure 19: Absolute 2-norm error of the approximation.

— ode45 — ode23 — ode113 — ode15s — ode23s — ode23t — ode23tb

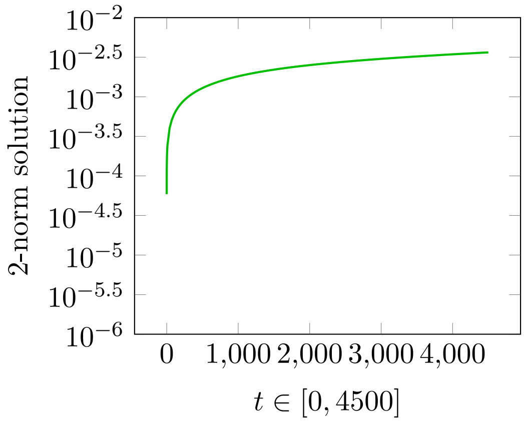


Figure 20: 2-norm of the reference solution.

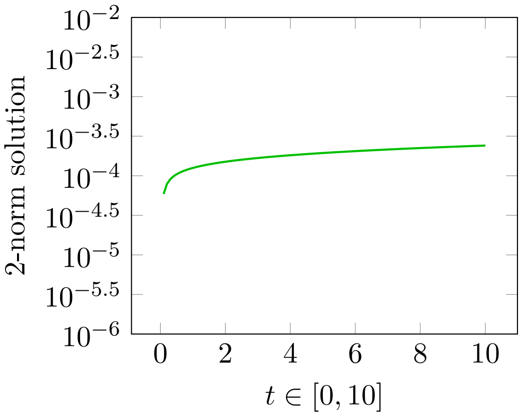


Figure 21: 2-norm of the reference solution.

A.2. Results for the Transposed Differential Lyapunov Equation

$$n = 1357 \text{ and } M^T\dot{X}(t)M = A^T X(t)M + M^T X(t)A + C^T C, \;\; X(0) = 0.$$

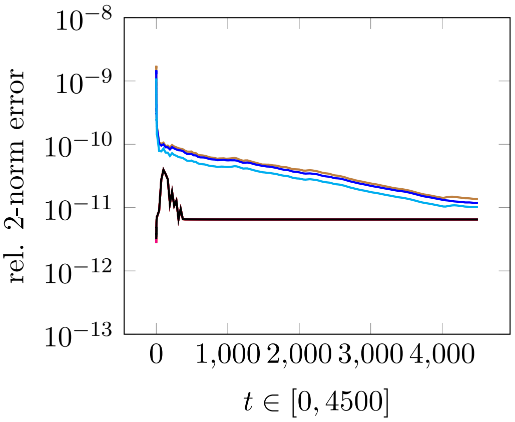


Figure 22: Relative 2-norm error of the approximation.

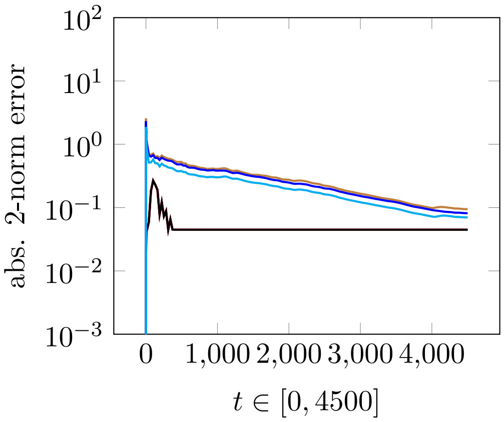


Figure 23: Absolute 2-norm error of the approximation.

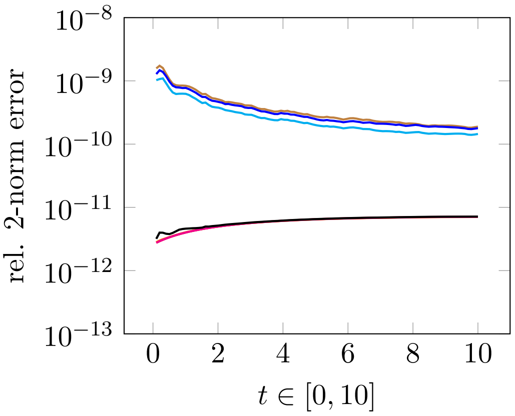


Figure 24: Relative 2-norm error of the approximation.

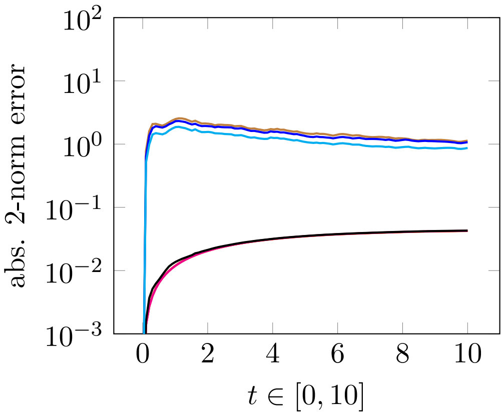


Figure 25: Absolute 2-norm error of the approximation.

— ode45 — ode23 — ode113 — ode15s — ode23s — ode23t — ode23tb

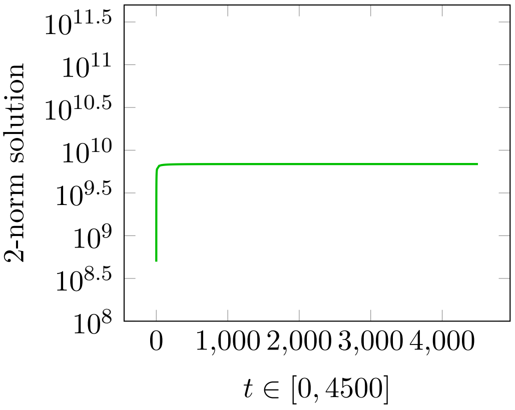


Figure 26: 2-norm of the reference solution.

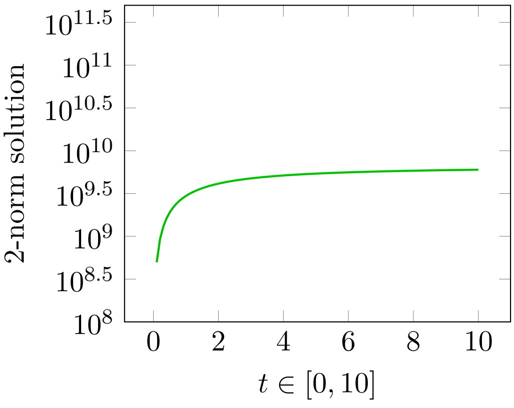


Figure 27: 2-norm of the reference solution.

$$n = 5177 \text{ and } M^T\dot{X}(t)M = A^TX(t)M + M^TX(t)A + C^TC,\;\; X(0) = 0.$$

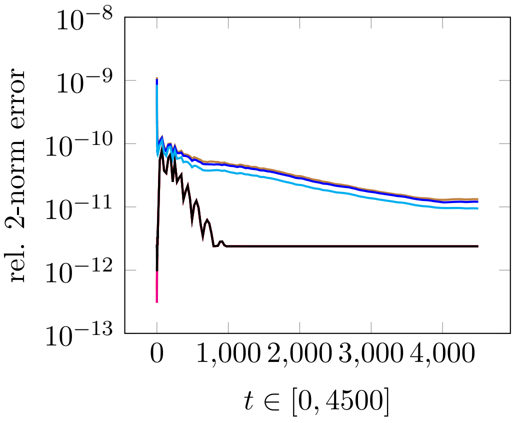


Figure 28: Relative 2-norm error of the approximation.

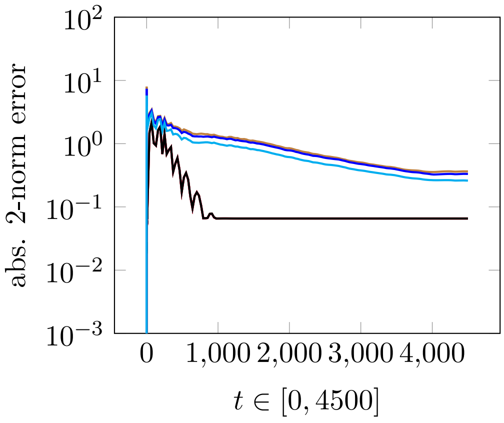


Figure 29: Absolute 2-norm error of the approximation.

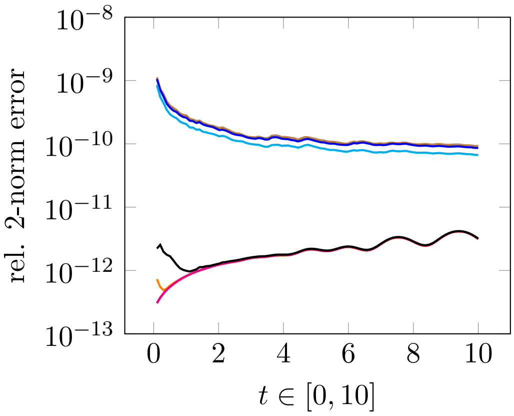


Figure 30: Relative 2-norm error of the approximation.

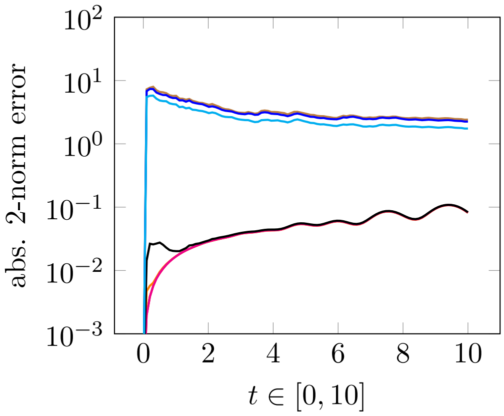


Figure 31: Absolute 2-norm error of the approximation.

— `ode45` — `ode23` — `ode113` — `ode15s` — `ode23s` — `ode23t` — `ode23tb`

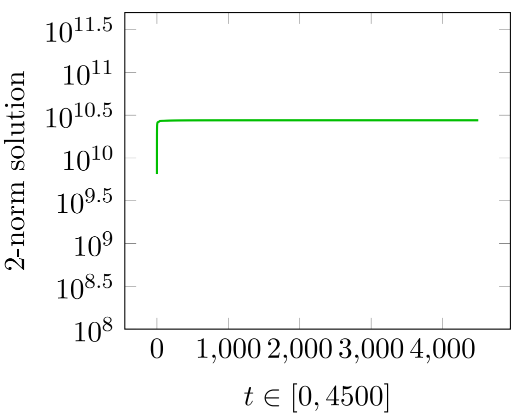


Figure 32: 2-norm of the reference solution.

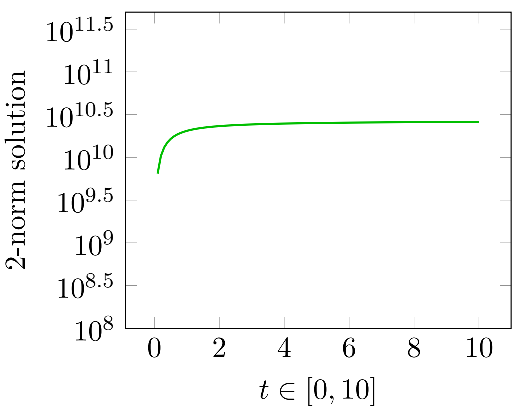


Figure 33: 2-norm of the reference solution.

$$n = 20209 \text{ and } M^T\dot{X}(t)M = A^TX(t)M + M^TX(t)A + C^TC,\ \ X(0) = 0.$$

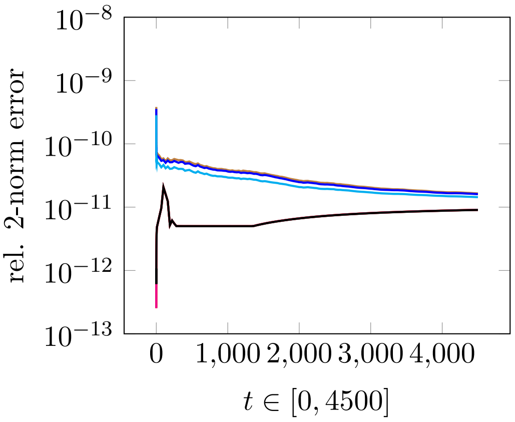


Figure 34: Relative 2-norm error of the approximation.

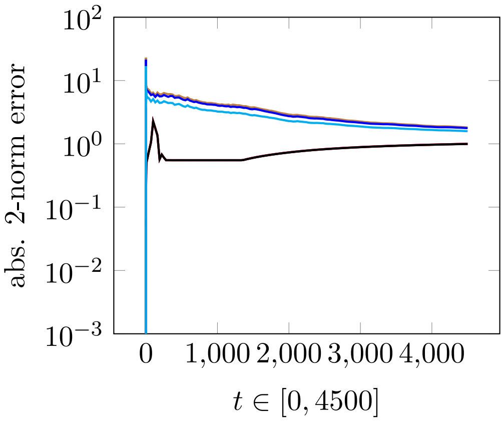


Figure 35: Absolute 2-norm error of the approximation.

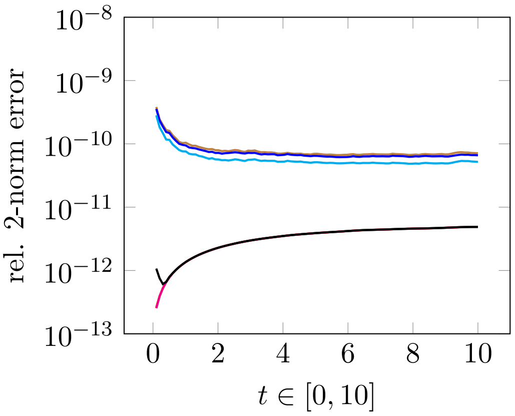


Figure 36: Relative 2-norm error of the approximation.

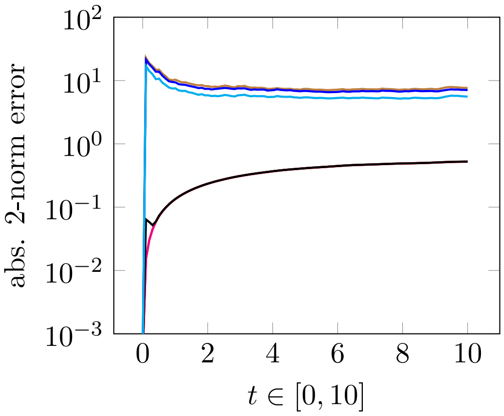


Figure 37: Absolute 2-norm error of the approximation.

— ode45 — ode23 — ode113 — ode15s — ode23s — ode23t — ode23tb

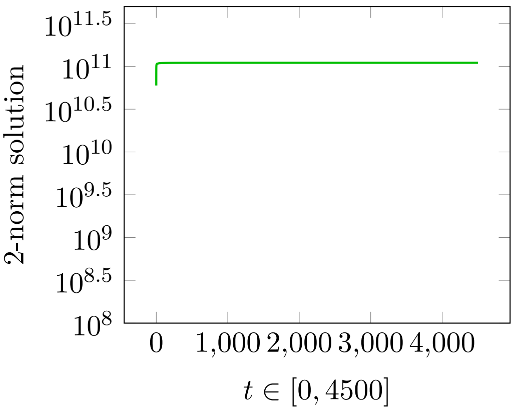


Figure 38: 2-norm of the reference solution.

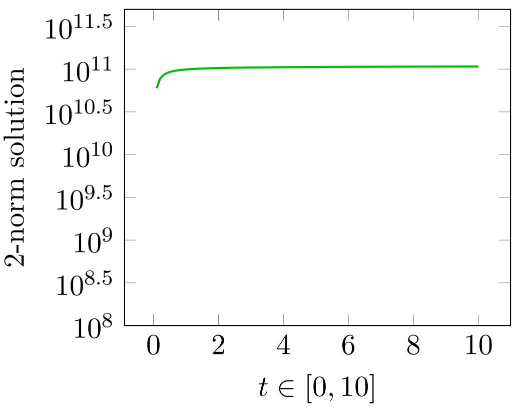


Figure 39: 2-norm of the reference solution.

B. Backward Differentiation Formulas

We consider Backward Differentiation Formulas (BDF) for differential Lyapunov equations [27,28]. Let $0 = t_0 < t_1 < \cdots < t_N = T$ be a decomposition of the interval $[0, T]$. We define the step size $\tau_k = t_k - t_{k-1}$ for $k = 1, \ldots, N$.

The s-step BDF method applied to the DLE 8 is given by

$$\sum_{j=0}^{s} \alpha_j X_{k-j} = \tau_k \beta \left(A^T X_k + X_k A + BB^T \right),$$

where α_j and β are coefficients of the BDF method and can be found in [15]. The parameter s is the order of the BDF method. We recall that for $s > 6$, the method is not numerical stable, and for $s = 1$, the BDF method coincides with the implicit Euler method. A minor rearrangement shows that the current iterate X_k can be obtained as the solution of the algebraic Lyapunov equation

$$\left(\tau_k \beta A - \frac{\alpha_0}{2} E_{n,n} \right)^T X_k + X_k \left(\tau_k \beta A - \frac{\alpha_0}{2} E_{n,n} \right) = -\tau_k \beta BB^T + \sum_{j=1}^{s} \alpha_j X_{k-j}. \tag{10}$$

Since for $s \geq 2$, certain coefficients α_j, $j \geq 1$ are positive, the algebraic Lyapunov equation (10) has a symmetric but possibly indefinite right-hand side, which makes the standard ADI method infeasible. For this reason a LDL^T-decomposition for the numerical solution is proposed and suitable modifications of the ADI method have been developed; [5, 26]: Assume that $X_i \approx L_i D_i L_i^T$ for $i = 0, \ldots, k-1$, $L_i \in \mathbb{R}^{n \times l_i}$, $D_i \in \mathbb{R}^{l_i \times l_i}$ and $l_i \ll n$, then the right hand side can be factored as

$$\begin{aligned} -\tau_k \beta BB^T + \sum_{j=1}^{s} \alpha_j X_{k-j} &\approx -G_k S_k G_k^T, \\ G_k &= \left[B, L_{k-1}, \ldots, L_{k-s} \right], \\ S_k &= \begin{bmatrix} \tau_k \beta E_{p,p} & & & \\ & -\alpha_1 D_{k-1} & & \\ & & \ddots & \\ & & & -\alpha_s D_{k-s} \end{bmatrix}. \end{aligned}$$

Now the LDL^T-type ADI method can be used to determine $X_k \approx L_k D_k L_k^T$. The BDF/ADI methods can also be extended to generalized differential Lyapunov equations in a similar way [28].

B.1. Results for the Differential Lyapunov Equation

$$n = 1357 \text{ and } M\dot{X}(t)M^T = AX(t)M^T + MX(t)A^T + BB^T,\ X(0) = 0.$$

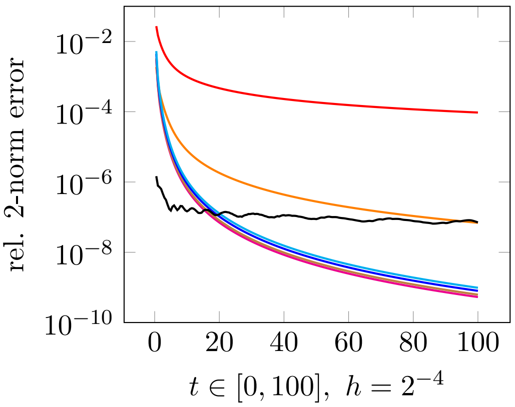


Figure 40: Relative 2-norm error of the BDF/ADI approximation.

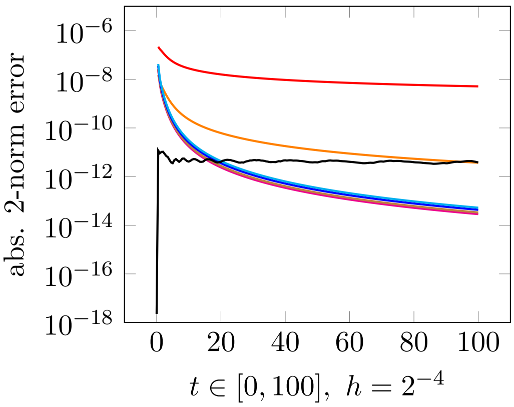


Figure 41: Absolute 2-norm error of the BDF/ADI approximation.

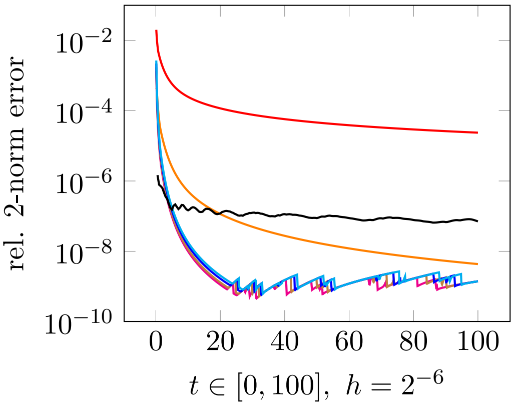


Figure 42: Relative 2-norm error of the BDF/ADI approximation.

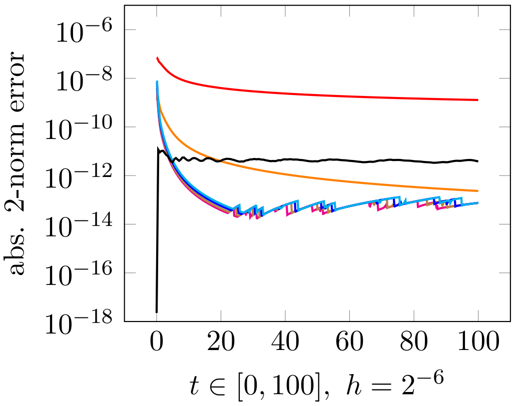


Figure 43: Absolute 2-norm error of the BDF/ADI approximation.

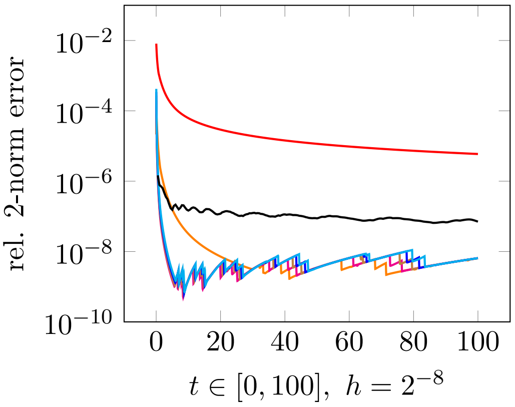


Figure 44: Relative 2-norm error of the BDF/ADI approximation.

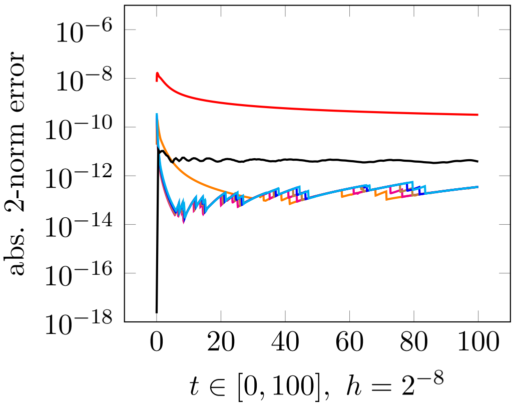


Figure 45: Absolute 2-norm error of the BDF/ADI approximation.

— `ode15s` — BDF1 — BDF2 — BDF3 — BDF4 — BDF5 — BDF6

B.2. Results for the Transposed Differential Lyapunov Equation

$$n = 1357 \text{ and } M^T\dot{X}(t)M = A^TX(t)M + M^TX(t)A + C^TC,\ \ X(0) = 0.$$

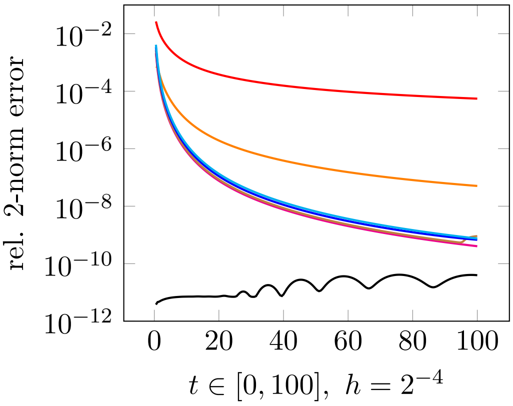


Figure 46: Relative 2-norm error of the BDF/ADI approximation.

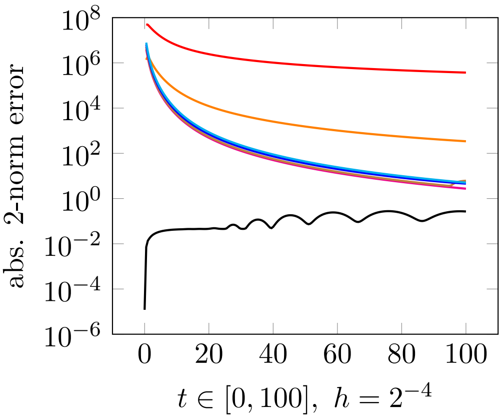


Figure 47: Absolute 2-norm error of the BDF/ADI approximation.

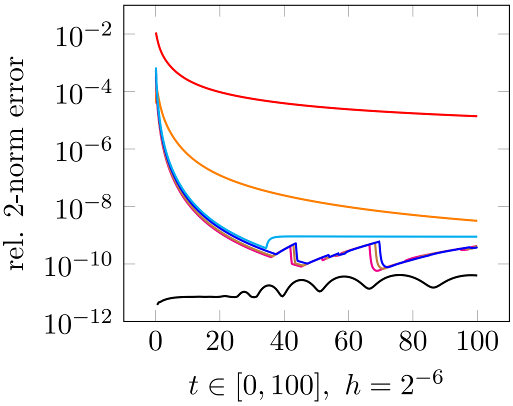


Figure 48: Relative 2-norm error of the BDF/ADI approximation.

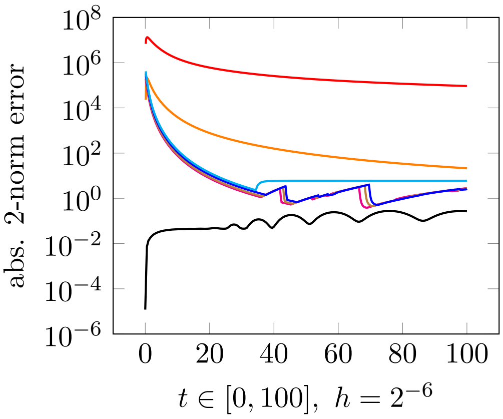


Figure 49: Absolute 2-norm error of the BDF/ADI approximation.

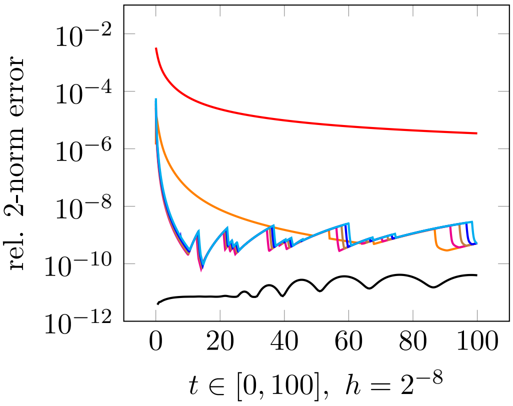


Figure 50: Relative 2-norm error of the BDF/ADI approximation.

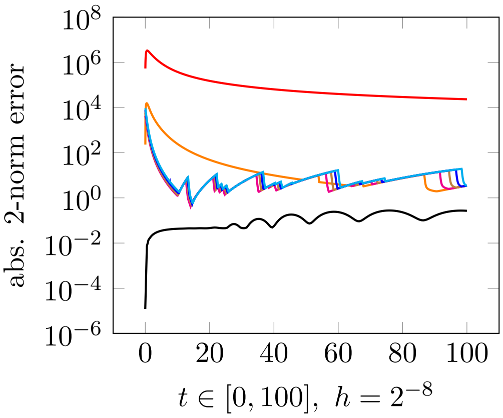


Figure 51: Absolute 2-norm error of the BDF/ADI approximation.

— `ode15s` — `BDF1` — `BDF2` — `BDF3` — `BDF4` — `BDF5` — `BDF6`

References

[1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank. *Matrix Riccati Equations in Control and Systems Theory.* Birkhäuser, Basel, Switzerland, 2003.

[2] F. Amato, R. Ambrosino, M. Ariola, C. Cosentino, and G. De Tommasi. *Finite-time stability and control*, volume 453 of *Lecture Notes in Control and Inform. Sci.* Springer, London, 2014.

[3] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*, volume 6 of *Adv. Des. Control.* SIAM Publications, Philadelphia, PA, 2005.

[4] P. Benner, M. Köhler, and J. Saak. M.E.S.S. – the matrix equations sparse solvers library. `https://www.mpi-magdeburg.mpg.de/projects/mess`.

[5] P. Benner, R.-C. Li, and N. Truhar. On the ADI method for Sylvester equations. *J. Comput. Appl. Math.*, 233(4):1035–1045, Dec. 2009.

[6] P. Benner and H. Mena. BDF methods for large-scale differential Riccati equations. In B. De Moor, B. Motmans, J. Willems, P. Van Dooren, and V. Blondel, editors, *Proc. 16th Intl. Symp. Mathematical Theory of Network and Systems, MTNS 2004*, 2004.

[7] P. Benner and H. Mena. Rosenbrock methods for solving Riccati differential equations. *IEEE Trans. Autom. Control*, 58(11):2950–2957, 2013.

[8] P. Benner and J. Saak. A semi-discretized heat transfer model for optimal cooling of steel profiles. In P. Benner, V. Mehrmann, and D. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lect. Notes Comput. Sci. Eng.*, pages 353–356. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

[9] R. A. Brocket. *Finite Dimensional Linear Systems.* Wiley, New York, 1970.

[10] R. Byers and S. Nash. On the singular "vectors" of the Lyapunov operator. *SIAM J. Algebraic Discrete Methods*, 8(1):59–66, 1987.

[11] Z. Gajić and M. Qureshi. *Lyapunov Matrix Equation in System Stability and Control.* Math. in Science and Engineering. Academic Press, San Diego, CA, 1995.

[12] Y. Güldoğan, M. Hached, K. Jbilou, and M. Kurulay. Low rank approximate solutions to large-scale differential matrix Riccati equations. Technical Report arXiv:1612.00499v2, arXiv, Apr. 2017. math.NA.

[13] M. Hached and K. Jbilou. Numerical solutions to large-scale differential Lyapunov matrix equations. *Numer. Algorithms*, Dec. 2017. Springer online first.

[14] M. Hached and K. Jbilou. Approximate solutions to large nonsymmetric differential Riccati problems. Technical Report arXiv:1801.01291v1, arXiv, Jan. 2018. math.NA.

[15] E. Hairer and G. Wanner. *Solving ordinary differential equations. I.* Springer Series in Computational Mathematics. Springer-Verlag, Berlin, 1987.

[16] J. Heiland. *Decoupling and Optimization of Differential-Algebraic Equations with Application in Flow Control.* Dissertation, TU Berlin, 2014.

[17] N. J. Higham. *Functions of matrices: Theory and computation.* Applied Mathematics. SIAM Publications, Philadelphia, PA, 2008.

[18] A. Jameson. Solution of the equation $AX + XB = C$ by inversion of an $M \times M$ or $N \times N$ matrix. *SIAM J. Appl. Math.*, 16:1020–1023, 1968.

[19] H. W. Knobloch and H. Kwakernaak. *Lineare Kontrolltheorie.* Springer-Verlag, Berlin, 1985. In German.

[20] L. Kohaupt. Solution of the matrix eigenvalue problem $VA + A^*V = \mu V$ with applications to the study of free linear dynamical systems. *J. Comput. Appl. Math.*, 213(1):142–165, 2008.

[21] M. Köhler, N. Lang, and J. Saak. Solving differential matrix equations using Parareal. *Proc. Appl. Math. Mech.*, 16(1):847–848, 2016.

[22] M. Konstantinov, V. Mehrmann, and P. Petkov. On properties of Sylvester and Lyapunov operators. *Linear Algebra Appl.*, 312(1-3):35–71, 2000.

[23] M. M. Konstantinov, D. W. Gu, V. Mehrmann, and P. H. Petkov. *Perturbation Theory for Matrix Equations*, volume 9 of *Stud. Comput. Math.* Elsevier, Amsterdam, 1st edition edition, May 2003.

[24] A. Koskela and H. Mena. A structure preserving Krylov subspace method for large scale differential Riccati equations. e-print arXiv:1705.07507, arXiv, May 2017. math.NA.

[25] N. Lang. *Numerical Methods for Large-Scale Linear Time-Varying Control Systems and related Differential Matrix Equations.* Dissertation, Technische Universität Chemnitz, Germany, June 2017.

[26] N. Lang, H. Mena, and J. Saak. On the benefits of the LDL^T factorization for large-scale differential matrix equation solvers. *Linear Algebra Appl.*, 480:44–71, 2015.

[27] N. Lang, J. Saak, and T. Stykel. Towards practical implementations of balanced truncation for LTV systems. *IFAC-PapersOnLine*, 48(1):7–8, 2015.

[28] N. Lang, J. Saak, and T. Stykel. Balanced truncation model reduction for linear time-varying systems. *Math. Comput. Model. Dyn. Syst.*, 22(4):267–281, 2016.

[29] N. Lang, J. Saak, and T. Stykel. LTV-BT for MATLAB. `https://doi.org/10.5281/zenodo.834953`, 2017.

[30] A. Locatelli. *Optimal Control: An Introduction.* Birkhäuser, Basel, Switzerland, 2001.

[31] H. Mena. *Numerical Solution of Differential Riccati Equations Arising in Optimal Control of Partial Differential Equations.* Dissertation, Escuela Politécnica Nacional, Ecuador, July 2007.

[32] T. W. Palmer. *Banach algebras and the general theory of $*$-algebras. Vol. 2*, volume 79 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, Cambridge, 2001. $*$-algebras.

[33] J. W. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory*, volume 26 of *Texts in Applied Mathematics.* Springer New York, 1998.

[34] H. Rome. A direct solution to the linear variance equation of a time-invariant linear system. *IEEE Trans. Autom. Control*, 14(5):592–593, 1969.

[35] G. W. Stewart. *Matrix algorithms. Vol. II.* SIAM Publications, Philadelphia, PA, 2001. Eigensystems.

[36] T. Stillfjord. Low-rank second-order splitting of large-scale differential Riccati equations. *IEEE Trans. Autom. Control*, 60(10):2791–2796, 2015.

[37] D. Werner. *Funktionalanalysis.* Springer-Verlag, Berlin, extended edition, 2000.