Iterative methods for the delay Lyapunov equation with T-Sylvester preconditioning

Elias Jarlebring[a], Federico Poloni[b]

[a] *Department of Mathematics, Royal Institute of Technology (KTH), Stockholm, SeRC Swedish e-Science Research Center.* eliasj@kth.se
[b] *Department of Computer Science, University of Pisa, Italy.* federico.poloni@unipi.it

Abstract

The delay Lyapunov equation is an important matrix boundary-value problem which arises as an analogue of the Lyapunov equation in the study of time-delay systems $\dot{x}(t) = A_0x(t) + A_1x(t-\tau) + B_0u(t)$. We propose a new algorithm for the solution of the delay Lyapunov equation. Our method is based on the fact that the delay Lyapunov equation can be expressed as a linear system of equations, whose unknown is the value $U(\tau/2) \in \mathbb{R}^{n\times n}$, i.e., the delay Lyapunov matrix at time $\tau/2$. This linear matrix equation with n^2 unknowns is solved by adapting a preconditioned iterative method such as GMRES. The action of the $n^2 \times n^2$ matrix associated to this linear system can be computed by solving a coupled matrix initial-value problem. A preconditioner for the iterative method is proposed based on solving a T-Sylvester equation $MX + X^TN = C$, for which there are methods available in the literature. We prove that the preconditioner is effective under certain assumptions. The efficiency of the approach is illustrated by applying it to a time-delay system stemming from the discretization of a partial differential equation with delay. Approximate solutions to this problem can be obtained for problems of size up to $n \approx 1000$, i.e., a linear system with $n^2 \approx 10^6$ unknowns, a dimension which is outside of the capabilities of the other existing methods for the delay Lyapunov equation.

Keywords: Matrix equations, iterative methods, Krylov methods, time-delay systems, Sylvester equations, ordinary differential equations

1. Introduction

Consider the linear single-delay time-delay system defined by the equations

$$\dot{x}(t) = A_0x(t) + A_1x(t-\tau) + B_0u(t) \tag{1a}$$
$$y(t) = C_0x(t), \tag{1b}$$

where $A_0, A_1 \in \mathbb{R}^{n\times n}$, $B_0 \in \mathbb{R}^{n\times m}$, $C_0^{\mathrm{T}} \in \mathbb{R}^{n\times p}$. The general equation (1) appears in many different fields. It is considered a very important topic in the field of systems and control, mostly due to the fact that most feedback systems

are non-instantaneous in the sense that there is a delay between the observation (of for instance the state) and the action of the feedback. See monographs [1, 2] and survey paper [3] for literature on time-delay systems.

The delay Lyapunov equations associated with (1) corresponds to the problem of finding $U \in \mathcal{C}^0([-\tau, \tau], \mathbb{C}^{n\times n})$ such that

$$U'(t) = U(t)A_0 + U(t-\tau)A_1,\ t > 0, \tag{2a}$$
$$U(-t) = U(t)^{\mathrm{T}}, \tag{2b}$$
$$-W = U(0)A_0 + A_0^{\mathrm{T}}U(0) + U(\tau)^{\mathrm{T}}A_1 + A_1^{\mathrm{T}}U(\tau), \tag{2c}$$

hold for a given a cost matrix $W = W^T \in \mathbb{R}^{n\times n}$ (in some applications, for instance, $W = C_0^T C_0$).

Equation (2a) is a matrix delay-differential equation and (2c) is an algebraic condition involving $U(0)$, $U(\tau)$ and $U(-\tau) = U(\tau)^T$ such that (2) can be interpreted as a matrix boundary value problem. In this paper we propose a new procedure to solve (2), with the goal to have good performance for large n.

The delay Lyapunov equation generalizes the standard Lyapunov equation, since e.g., if we set $\tau = 0$ the equation reduces to the standard Lyapunov equation. Moreover, as established by the last decades of research, the delay Lyapunov equation is in many ways playing the same important role for time-delay systems as the standard Lyapunov equation plays for standard (delay free) linear time-invariant dynamical systems. More precisely, the delay Lyapunov equation has been studied in the following ways. It has been extensively used to characterize stability of delay differential equations, as one can explicitly construct a Lyapunov functional from $U(t)$, where the solution is sometimes referred to as delay Lyapunov matrices. Sufficient conditions for stability are given in [4] [5], [6] and for neutral systems in [7], and conditions for instability in [8, 9]. It has been used to provide bounds on the transient phase of delay-differential equations in the PhD thesis [10] and [11, 12]. Existence and uniqueness of the solutions are well characterized, e.g., in [4]. See also the monograph [2]. Recently, it has been shown that in complete analogy to the standard Lyapunov equation the solution to the delay Lyapunov equation explicitly gives the $\mathcal{H}_2$-norm [13]. The delay Lyapunov equation can also be used to carry out a model order reduction which generalizes balanced truncation [14].

This paper concerns computational aspects of the delay Lyapunov equation. Some computational aspects are treated in the literature, e.g., the matrix exponential formula in [10], the polynomial approximation approach in [15], spectral (Chebyshev-based) discretization approaches in [13, 16] and an ODE-approach in the PhD thesis [17, Chapter 3].

In complete contrast to the delay Lyapunov equation, the computational aspects of the standard Lyapunov equation have received considerable attention, mostly in the numerical linear algebra community. Most importantly, the Bartels-Stewart method [18], ADI methods [19], Krylov methods [20, 21], and rational Krylov methods [22], including preconditioning techniques [23], have turned to be effective in various situations. For a more thorough review, see the survey [24]. To our knowledge, there exist no natural generalization of the

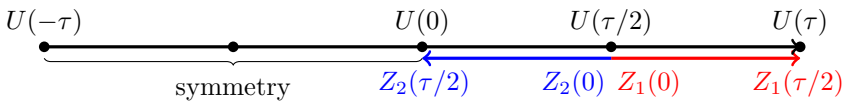


Figure 1: Graphical representation of the relation between $U(t)$, $Z_1(t)$ and $Z_2(t)$.

Bartels-Stewart algorithm and there are no Krylov methods for delay Lyapunov equation.

The method we propose is tailored to medium-scale equations; it combines the use of a Krylov-type method and a direct algorithm similar to the Bartels-Stewart one. More precisely, our approach is based on a characterization of the solution to the delay Lyapunov equation as a linear system of equations with n^2 unknowns. This characterization is derived in Section 2. Since the linear system derived in Section 2 is large and only given implicitly as a matrix vector product, we propose to adapt iterative methods which are based on matrix vector products only, e.g., GMRES [25] or BiCGStab [26], to this problem. It turns out to be natural to use a preconditioner involving a matrix equation called the T-Sylvester equation, for which there are efficient $O(n^3)$ methods for the dense case [27]. We quantify the quality of the preconditioner by deriving a bound on the convergence factor of the iterative method. The iterative method and the preconditioner are given in Section 3. The performance of the approach is illustrated with simulations in Section 4. We apply the method to a problem stemming from the discretization of a two-dimensional partial delay-differential equation (PDDE). The number of iterations appears to be essentially independent of the grid, which suggests that the preconditioner is a sensible choice for this PDDE.

We use notation which is standard for analysis of matrix equations. The vectorization operation is denoted $\mathrm{vec}(B)$, i.e., if $B = \begin{bmatrix} b_1 & \dots & b_m \end{bmatrix} \in \mathbb{R}^{n\times m}$, $\mathrm{vec}(B)^T = \begin{bmatrix} b_1^T & \dots & b_m^T \end{bmatrix}$. The Kronecker product is denoted $\otimes$. Unless otherwise stated, $\|\cdot\|$ denotes the Euclidean norm for vectors and the spectral norm for matrices. We denote the Frobenius norm by $\|\cdot\|_F$.

2. Reformulation of the delay Lyapunov equations

Our method is based on a reformulation of the delay Lyapunov equation where we define for each $t \in [0, \tau/2]$

$$Z_1(t) := U(\tau/2 + t), \qquad Z_2(t) := U(\tau/2 - t). \tag{3}$$

The two matrix-valued functions $Z_1(t)$ and $Z_2(t)$ coincide with $U(t)$ up to a change of the time coordinate which is represented visually in Figure 1. Essentially, they represent two different branches of $U(t)$ "taking off" from $\tau/2$ in opposite directions. Note that the left half of the function, $U([-\tau, 0])$, is determined uniquely by the right half $U([0, \tau])$ by the transposition symmetry

condition (2b). The only nontrivial condition implied by (2b) is that $U(0)$ must be symmetric.

Note that

$$\begin{aligned}
(4a)\, Z_1(t-\tau) &= U(t-\tau+\tau/2) = U(t-\tau/2) = U(\tau/2-t)^T = Z_2(t)^T \\
(4b)\, Z_2(t-\tau) &= U(\tau/2-t-\tau) = U(-t-\tau/2) = U(t+\tau/2)^T = Z_1^T(t)
\end{aligned}$$

Hence, the delay differential equation (2a) becomes an ordinary differential equation

$$\begin{aligned}
(5a) \qquad Z_1'(t) &= Z_1(t)A_0 + Z_2(t)^T A_1, \\
(5b) \qquad Z_2'(t) &= -Z_1(t)^T A_1 - Z_2(t)A_0.
\end{aligned}$$

This is a constant-coefficient homogeneous linear system of ODEs which can be solved explicitly if the common (unknown) initial value $Z_1(0) = Z_2(0) = U(\tau/2)$ is provided. Using vectorization, we can give an explicit formula

$$(6) \qquad \begin{bmatrix} \operatorname{vec} Z_1(t) \\ \operatorname{vec} Z_2(t)^T \end{bmatrix} = \exp(t\mathcal{A}) \begin{bmatrix} \operatorname{vec} U(\tau/2) \\ \operatorname{vec} U(\tau/2)^T \end{bmatrix},$$

where

$$(7) \qquad \mathcal{A} := \begin{bmatrix} A_0^T \otimes I_n & A_1^T \otimes I_n \\ -I_n \otimes A_1^T & -I_n \otimes A_0^T \end{bmatrix}.$$

In terms of $Z_1(t)$ and $Z_2(t)$, the algebraic condition (2c) and the symmetry condition (2b) for $t = 0$ reduce to

$$\begin{aligned}
(8a)\ 0 &= W + Z_2(\tau/2)^T A_0 + A_0^T Z_2(\tau/2) + Z_1(\tau/2)^T A_1 + A_1^T Z_1(\tau/2), \\
(8b)\ 0 &= Z_2(\tau/2) - Z_2(\tau/2)^T.
\end{aligned}$$

Notice that the right-hand side of (8a) is symmetric and that of (8b) is antisymmetric. A linear combination of them gives

$$(9)\ \ 0 = W + Z_2(\tau/2)^T(A_0 - cI) + (A_0^T + cI)Z_2(\tau/2) + Z_1(\tau/2)^T A_1 + A_1^T Z_1(\tau/2)$$

for each $c \in \mathbb{R}$, which forms the basis of our matrix operator.

Definition 1. *Let $L_c : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ be defined by*

$$\begin{aligned}
(10) \quad L_c(X) := \\
Z_2(\tau/2)^T(A_0 - cI) + (A_0^T + cI)Z_2(\tau/2) + Z_1(\tau/2)^T A_1 + A_1^T Z_1(\tau/2)
\end{aligned}$$

where $Z_i : [0, \tau/2] \to \mathbb{R}^{n\times n}$, $i = 1, 2$ are the unique solutions to the initial value problem (5) with $Z_1(0) = Z_2(0) = X$.

We shall need the following easy linear algebra result.

Lemma 2. *Let $M = M^T \in \mathbb{R}^{n\times n}$ and $N = -N^T \in \mathbb{R}^{n\times n}$ be two matrices, one symmetric and one antisymmetric. Then, $M+N = 0$ if and only if $M = N = 0$.*

Proof. The 'if' part is trivial; let us prove the 'only if'. Suppose $M + N = 0$; then, transposing, we have also $0 = M^T + N^T = M - N$. Summing and subtracting the two relations we have $2M = 2N = 0$. □

A time-delay system is called *exponentially stable* if $\|x(t)\| \leq M \exp(-\beta t)$ for some constants $M > 0, \beta > 0$. If this condition holds, then the solution $U(t)$ to (2) is unique [12, Theorem 4]. In this case, we can formulate the equivalence between the delay Lyapunov equation and a linear system with operator L_c.

Theorem 3 (Equivalence)**.** *Suppose A_0 and A_1 and τ are such that* (1) *is exponentially stable and let $W \in \mathbb{R}^{n\times n}$ be any symmetric matrix. Let U be the solution to the delay Lyapunov matrices* (2) *and let L_c be defined by* (10)*. Then, for any $c \neq 0$, $X = U(\tau/2)$ is the unique solution of the linear system*

$$L_c(X) = -W. \tag{11}$$

Proof. Equation (9) already shows that if $X = U(\tau/2)$ then $L_c(X) = -W$. It remains to prove the reverse implication. Suppose that X satisfies $L_c(X)+W = 0$; then, by Lemma 2 applied to

$$\begin{aligned} M &= Z_2(\tau/2)^T A_0 + A_0^T Z_2(\tau/2) + Z_1(\tau/2)^T A_1 + A_1^T Z_1(\tau/2) - W, \\ N &= c(Z_2(\tau/2) - Z_2(\tau/2)^T), \end{aligned}$$

the conditions (8) hold. Define

$$\hat{U}(t) = \begin{cases} Z_2(\tau/2 - t) & 0 \leq t < \tau/2, \\ Z_1(t - \tau/2) & \tau/2 \leq t \leq \tau, \\ U(-t)^T & -\tau \leq t < 0. \end{cases}$$

The function $\hat{U}(t)$ is continuous in 0 by (8b), and in $\pm\tau/2$ by the choice of initial conditions, hence it is globally continuous on $[-\tau, \tau]$. Moreover, the differential equation (2a) holds for all $t \neq 0, \tau/2$. By continuity, it must also hold for these values. Hence $\hat{U}(t)$ solves (2). As we assume exponential stability, the solution is unique and hence $\hat{U}(t) = U(t)$. □

Since the linear system $L_c(X) = -W$ has a unique solution for each symmetric $W \in \mathbb{R}^{n\times n}$, we have the following result.

Corollary 4. *Suppose* (1) *is exponentially stable. Then, the linear operator L_c is nonsingular for each $c \neq 0$.*

A delay-free formulation of the delay Lyapunov equations has also been derived in [4, Equation (13)]. That formulation cannot be described with a linear operator in a way that can be adapted to an iterative method in the same way that we show in the following section.

3. Algorithm

We now know from the previous section that the matrix equation (11) is equivalent to the delay Lyapunov equation. By vectorizing (11), we obtain the linear system on standard form

$$\operatorname{vec} L_c(\operatorname{vec}^{-1} x) = -\operatorname{vec} W, \tag{12}$$

where the inverse function $\operatorname{vec}^{-1}(x)$ maps $\operatorname{vec} X \in \mathbb{R}^{n^2}$ to $X \in \mathbb{R}^{n\times n}$. Let $A \in \mathbb{R}^{n^2\times n^2}$ the matrix associated to it. We know that A is nonsingular by Corollary 4.

Our approach is based on specializing an iterative method for linear systems to (12). In order to specialize an iterative method for large-scale linear systems, we need two ingredients. We need an efficient procedure to compute the action corresponding to the left-hand side of (12); and we need a preconditioner. These two ingredients are described in the following two subsections.

3.1. Action of L_c

The action of the operator L_c is defined by (5) and (10). As a consequence, the recipe to compute $L_c(X)$ for a given matrix X is simple:

1. Compute the solutions $Z_1(\tau/2)$, $Z_2(\tau/2)$ of the linear, constant-coefficient initial-value problem (5) with initial values $Z_1(0) = Z_2(0) = X$.
2. Compute $L_c(X)$ using the expression (10).

In practice, a detail is crucial in the choice of the numerical algorithm for the first step. We distinguish two possible scenarios:

- We use a method with a fixed step-size and no adaptivity: for instance, the (explicit or implicit) Euler method, or a non-adaptive Runge-Kutta method. In this case, we are effectively substituting L_c with a different operator $\hat{L}_c$, which replaces the differential operator in Step 1 with a finite discretization. This operator (for most classical methods) is still linear, so the theory of Krylov subspace methods can be applied without changes: we are applying a Krylov method to get an approximate solution of a nearby linear problem $\hat{L}_c$.

- We use an adaptive method, which can change step size along the algorithm, possibly in different ways for different initial values X. For instance, the Runge-Kutta-Fehlberg method (Matlab's `ode45`). While apparently the two cases are similar, the addition of adaptivity has an important consequence: the computed operator $\hat{L}_c$, this time, is no longer a linear operator, because in general $\hat{L}_c(X_1 + X_2) \neq \hat{L}_c(X_1) + \hat{L}_c(X_2)$. Indeed, for different values of the input X the initial-value problems could be solved using different grids, and hence different discrete approximations of the propagation operator. Thus we are in the realm of inexact Krylov methods [28]. The theory in this case is more involved, as more care is required with the error thresholds.

It turns out in the simulations (in Section 4) that it is advantageous to try to keep the number of iterations as low as possible. In this paper we therefore focus on the first approach, since an inexact Krylov method can require more iterations.

3.2. Preconditioning

In order to make iterative methods effective, it is common to carry out a transformation which preconditions the problem. This can often be interpreted as transforming the problem with an approximation of the inverse of the matrix/operator. We focus on a particular preconditioner obtained by solving the problem exactly when A_1 is replaced with the zero matrix. Then (10) becomes

$$\tilde{L}_c(X) := Z_2(\tau/2)^T(A_0 - cI) + (A_0^T + cI)Z_2(\tau/2), \tag{13}$$

and (5b) decouples from Z_1 such that $Z_2' = -Z_2(t)A_0$, which we can solve explicitly to get $Z_2(\tau/2) = X\exp(-\tau A_0/2)$.

Let T be the operator

$$T(Y) = (A_0^T + cI)Y + Y^T(A_0 - cI).$$

The operator L_c is invertible if and only T^{-1} exists, and in this case we have

$$\tilde{L}_c^{-1}(Z) = T^{-1}(Z)\exp(\tau A_0/2). \tag{14}$$

Inverting the operator T correspond to solving the so-called (real) *T-Sylvester equation* $MY + Y^TN = C$. The paper [27] discusses the solvability of this equation and presents a direct $O(n^3)$ Bartels–Stewart-like algorithm for its solution. In particular, the following result holds.

Theorem 5 ([29, Lemma 8],[27])**.** *Let $M, N, C \in \mathbb{R}^{n\times n}$. The equation $MX + X^TN = C$ has a unique solution X for each right-hand side C if and only if $\mu_i\bar{\mu}_j \neq 1$ for each pair μ_i, μ_j of eigenvalues of the pencil $M - \lambda N^T$.*

In our case, $M = A_0^T + cI$, $N = A_0 - cI$, so after a quick computation the solvability condition reduces to the following condition, which is independent of c.

Definition 6 (Hamiltonian eigenpairing)**.** *We say that the matrix $A_0 \in \mathbb{R}^{n\times n}$ has no* Hamiltonian eigenpairing*, if for each pair of eigenvalues λ_i, λ_j of the matrix A_0, we have*

$$\lambda_i + \bar{\lambda}_j \neq 0.$$

A matrix has no Hamiltonian eigenpairing, for instance, if $\Re\lambda < 0$ for each eigenvalue λ of A_0, i.e., if the delay-free system obtained by setting $A_1 = 0$ is stable.

In order to characterize the convergence and quality of the preconditioner we use a fundamental min-max bound. Suppose we carry out GMRES on the

matrix $A \in \mathbb{R}^{N\times N}$ with eigenvalues $\lambda_1, \ldots, \lambda_N$. From [25, Proposition 4] we have the bound of the residual

$$\|r_{m+1}\| \leq \kappa(X)\varepsilon^{(m)}\|r_0\|$$

and

$$\varepsilon^{(m)} = \min_{p\in P_m} \max_i |p(\lambda_i)|$$

where $P_m = \{p : \text{polynomial of degree } m \text{ such that } p(0) = 1\}$. We now apply the standard Zarantonello bound [30, Lemma 6.26], where we assume that the eigenvalues are contained in a disk of radius r centered at $c = 1$, corresponding to selecting $p(z) = \frac{(c-z)^m}{c^m}$ such that $\varepsilon^{(m)} \leq r^m/c^m = r^m \leq \|A - I\|^m$. Since preconditioned GMRES with preconditioner $\tilde{A}^{-1}$ is essentially equivalent to GMRES applied to the matrix $\tilde{A}^{-1}A$, a bound on $\|\tilde{A}^{-1}A - I\|$ provides a characterization of the convergence factor of preconditioned GMRES. Because of the vectorization included in our setting, bounding $\|\tilde{A}^{-1}A - I\|$ corresponds to giving an estimate for the quantity

$$\frac{\|\tilde{L}_c^{-1}(L_c(X)) - X\|_F}{\|X\|_F}.$$

Our preconditioner is constructed by setting $A_1 = 0$. Therefore, we expect that the preconditioner works well if $\|A_1\|$ is small. This reasoning is formalized in the following result.

Theorem 7 (Quality of preconditioner)**.** *Suppose the system* (1) *is exponentially stable and suppose that A_0 has no Hamiltonian eigenpairing. Let L_c and $\tilde{L}_c$ be defined by* (10) *and* (13) *respectively. Then,*

$$\frac{\|\tilde{L}_c^{-1}(L_c(X)) - X\|_F}{\|X\|_F} = \mathcal{O}(\|A_1\|_2), \tag{15}$$

where the constant hidden in the $\mathcal{O}(\cdot)$ notation depends only on $\|A_0\|$, τ and c.

Proof. We invoke Lemma 9 to bound the left-hand side of (15)

$$\frac{\left\|\tilde{L}_c^{-1}\left(L_c(X)\right) - X\right\|_F}{\|X\|_F} = \frac{\left\|\tilde{L}_c^{-1}\left(L_c(X) - \tilde{L}_c(X)\right)\right\|_F}{\|X\|_F} \leq K\exp(\tau\|A_0\|/2)\frac{\left\|L_c(X) - \tilde{L}_c(X)\right\|_F}{\|X\|_F}. \tag{16}$$

In order to bound $L_c(X) - \tilde{L}_c(X)$ we let Z_1 and Z_2 correspond to $L_c(X)$, i.e., they satisfy the equations (5) with initial value $Z_1(0) = Z_2(0) = X$. Let $\tilde{Z}_1$ and $\tilde{Z}_2$ be the corresponding quantities for $\tilde{L}_c(X)$. Moreover, let $\Delta_2 := Z_2 - \tilde{Z}_2$. We have

$$\begin{aligned}\tilde{L}_c(X) - L_c(X) = \\ \Delta_2(\tau/2)^T(A_0 - cI) + (A_0^T + cI)\Delta_2(\tau/2) + Z_1(\tau/2)^T A_1 + A_1^T Z_1(\tau/2),\end{aligned} \tag{17}$$

for which $\Delta_2(\tau/2)$ and $Z_1(\tau/2)$ can be bounded as follows. Lemma 8 tells us that

$$\|Z_1(\tau/2)\|_F \leq 2\exp(\tau(\|A_0\|_2 + \|A_1\|_2))\|X\|_F. \tag{18}$$

By definition, Δ_2 satisfies the ODE

$$\Delta_2'(t) = -\Delta_2(t)A_0 + g(t), \quad \Delta_2(0) = 0, \tag{19}$$

where $g(t) := -Z_1(t)^T A_1$. The variation-of-constants formula applied to (19) results in the explicit expression

$$\Delta_2(t) = -\int_0^t Z_1(s)^T A_1 \exp((s-t)A_0)\, ds.$$

Hence,

$$\|\Delta_2(\tau/2)\|_F \leq \int_0^{\tau/2} \|Z_1(s)^T A_1 \exp((s-\tau/2)A_0)\|_F\, ds \tag{20a}$$

$$\leq \int_0^{\tau/2} \|Z_1(s)\|_F \|A_1\|_2 \|\exp((s-\tau/2)A_0)\|_2\, ds \tag{20b}$$

$$\leq \tau\exp(\tau(\|A_0\|_2 + \|A_1\|_2))\|A_1\|_2 \exp(\tau\|A_0\|_2/2)\|X\|_F. \tag{20c}$$

We now evaluate the Frobenius norm of (17) and apply the triangle inequality and the bounds (18) and (20), which shows that

$$\frac{\|\tilde{L}_c(X) - L_c(X)\|_F}{\|X\|_F} = \mathcal{O}(\|A_1\|_2). \tag{21}$$

The hidden constant in (21) depends only on $\|A_0\|_2$, c, and τ. The conclusion (15) follows by combining (16) and (21). □

4. Simulations

4.1. A small example

In order to illustrate the preconditioner and properties of our approach we first consider a small example with randomly generated A_0 matrix. We specify the matrices for reproducibility

$$A_0 = \begin{bmatrix} -26 & 22 & -1 & -4 \\ 2 & -24 & -4 & 1 \\ 7 & 11 & -24 & -22 \\ -13 & 15 & -1 & -9 \end{bmatrix}, \quad A_1 = \alpha\,\mathrm{diag}(-1, -0.5, 0, 0.5), \quad W = I$$

and $\tau = 1$. We carry out simulations for different $\alpha = \|A_1\|$. The time-delay system is stable for all $\alpha \in [0, 10]$. The corresponding delay Lyapunov equation satisfies

$$U(\tau/2) \approx \frac{1}{100} \cdot \begin{bmatrix} 0.2302 & -0.0156 & 0.0101 & -0.3729 \\ -0.0885 & 0.0044 & -0.0038 & 0.1380 \\ 0.1466 & -0.0057 & 0.0056 & -0.2263 \\ -0.5485 & 0.0331 & -0.0238 & 0.8755 \end{bmatrix}$$

for $\alpha = 1$.

We combine our approach with two different generic iterative methods for linear systems of equations, GMRES [25] and BiCGStab [26] and select $c = 1$. To illustrate the properties of the performance of the iterative method, we solve the ODE defining L_c to full precision with the matrix exponential. The absolute error as a function of iteration is given in Figure 2. Both methods successfully solve the problem before the break-down at iteration n^2 except for $\|A_1\| = 10$. No substantial difference between the two iterative methods can be observed in the error as a function of iteration, i.e., nothing can be concluded regarding which of the two variants is better for this problem. The convergence of the two methods is faster for small $\|A_1\|$. This is due to the fact that the preconditioner is more effective when $\|A_1\|$ is small, which is consistent with Theorem 7 and Figure 3, where we clearly see that the norm of the preconditioned system $X \mapsto \tilde{L}_c^{-1}(L_c(X))$ has a linear dependence on $\|A_1\|$. The same conclusion is supported by the localization of the eigenvalues of the linear map $X \mapsto \tilde{L}_c^{-1}(L_c(X))$ in Figure 3b.

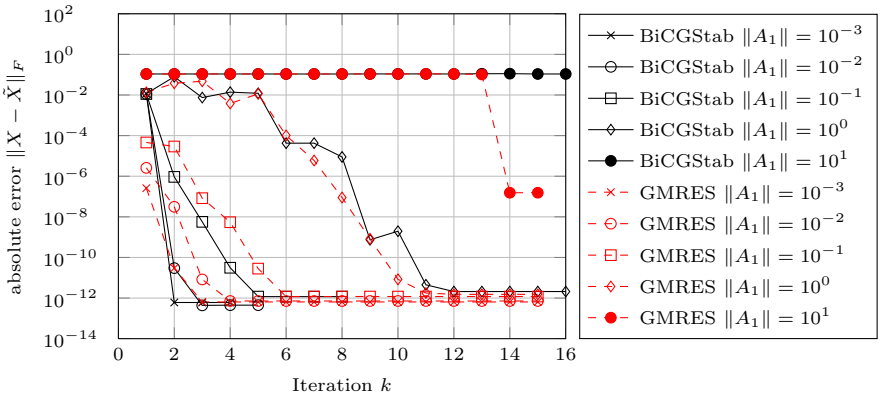
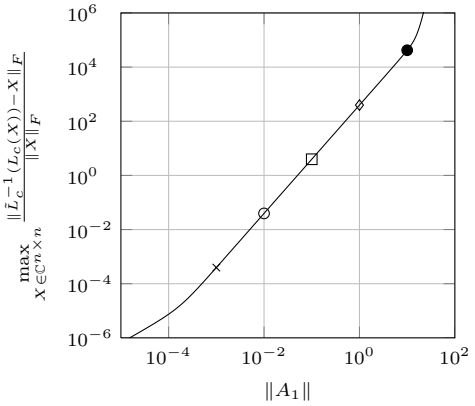


Figure 2: Convergence for different preconditioned iterative methods applied to the small example in Section 4.1

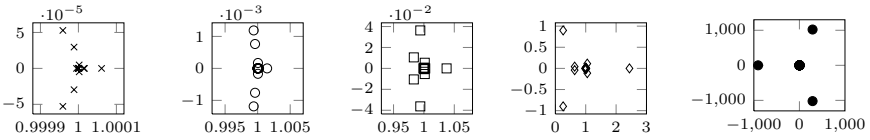
4.2. A large-scale example

In relation to other methods for delay Lyapunov equations, our iterative approach is likely to have better relative performance for large problems. We illustrate this with the following time-delay system stemming from the discretization of a partial differential equation with delay[1]. More precisely, we consider on the

[1]The MATLAB-code for the example and the simulation is publicly available on `http://www.math.kth.se/~eliasj/src/dlyap_precond`



(a) The preconditioner is better for small $\|A_1\|$ as predicted by Theorem 7



(b) Real and imaginary parts of the eigenvalues of the operator $X \mapsto \tilde{L}_c^{-1}(L_c(X))$ for different α with symbols specified in Figure 2.

Figure 3: Illustration of the quality of the preconditioner.

domain $(x, y) \in [0, 1] \times [0, 1]$ the PDDE

$$\text{(22a)} \quad \ddot{v}(x,y,t) = \Delta v(x,y,t) + \dot{v}(x,y,t) + f(x,z)\frac{\partial v}{\partial x}(x,y,t-\tau) + u(t)$$

$$\text{(22b)} \quad w(t) = v(1/2, 1/2)$$

where $f(x, y) = f_0 \cos(xy) \sin(\pi x)$ with homogeneous Dirichlet boundary conditions, and $f_0 = 5$. The PDDE (22) can be interpreted as waves propagating on a square, with damping and delayed feedback control. PDDEs are for instance studied in [31]. In order to reach a problem of the form (1) we rephrase (22) as a system of PDDEs which is first-order in time. We carry out a semi-discretization with finite differences in space with $n_x + 1$ intervals in the x-direction and $n_y + 1$ intervals in the y-direction, i.e., $h_x = 1/(n_x + 1)$, $x_k = kh_x$, $k = 1, \ldots, n_x$ and $h_y = 1/(n_y + 1)$, $y_k = kh_y$, $k = 1, \ldots, n_y$. The corresponding discretized

time-delay system is of the form (1) with coefficient matrices given by

$$A_0 = \begin{bmatrix} 0 & I \\ I \otimes D_{xx} + D_{yy} \otimes I & -I \end{bmatrix} \tag{23a}$$

$$A_1 = \begin{bmatrix} 0 & 0 \\ \mathrm{diag}(F)(I \otimes D_x) & 0 \end{bmatrix} \tag{23b}$$

$$B_0 = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix}^T \tag{23c}$$

$$C_0 = \begin{bmatrix} e_{(n_y+1)/2}^T \otimes e_{(n_x+1)/2}^T & 0 & \cdots & 0 \end{bmatrix} \tag{23d}$$

where

$$D_{xx} = \frac{1}{h_x^2} \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{n_x \times n_x}, \; D_{yy} = \frac{1}{h_y^2} \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{n_y \times n_y},$$

$$D_x = \frac{1}{2h_x} \begin{bmatrix} 0 & 1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & -1 & 0 \end{bmatrix} \in \mathbb{R}^{n_x \times n_x}, \quad F = \mathrm{vec}([f(x_i, y_j)]_{i,j=1}^{n_x, n_y}).$$

In the setting of $\mathcal{H}_2$-norm computation (as in [13]) we need to solve the delay Lyapunov equation with $W = C_0^T C_0$.

We carried out simulations of this system using a computer with an Intel i7 quad-core processor with 2.1GHz and 16 GB of RAM. For the finest discretization that we could treat with our approach, we have $n_x = n_y = 23$, $n = 1058$, $\|A_0\|_2 \approx 5000$ and $\|A_1\| \approx 100$. We again select $c = 1$.

In order to solve the ODE (5) we used a fixed fourth order Runge-Kutta method with $N = 500$ grid points. The iteration history of the two variants is visualized in Figure 4 for $n = 1058$. We observe linear convergence and no substantial difference in convergence rate. The execution time of our approach, in relation to some other approaches in the literature are reported in Table 1. We observe better relative computation time for larger problems. Moreover, we observe that other approaches fail due to requirement on memory resources.

Note also in Table 1 that the number of iterations required to reach a specified tolerance appears not to grow substantially with the size of problem. Hence, the method appears to have essentially grid-independent convergence rate, which is considered a very important feature of a preconditioner.

In a detailed profiling of our approach, we identify that two components are dominating, solving the ODE, i.e., computing the action, and solving the T-Sylvester equation. For the finest discretization, solving one T-Sylvester equation took approximately 320 seconds and carrying out one step of the ODE required 30 seconds. Since the main computational effort lies on the solution of the T-Sylvester equation and not in the computation of the action of L_c, it is for this problem no advantage to use adaptivity (as discussed in Section 3.1). In fact, the number of iterations can even increase when inexact Krylov methods are used.

We note that the implementation that we have used to solve T-Sylvester equations is not particularly optimized; it is a vectorized version of the algorithm in [27] that we have implemented in MATLAB for use in these experiments. The complexity in flops of the required computations is only slightly larger than what is required for solving a standard Sylvester equation with the Bartels-Stewart algorithm, a task which requires less than 8 seconds on our machine. Hence, we expect a major reduction in the timings if a carefully optimized solver for the T-Sylvester is used instead.

To our knowledge, the largest delay Lyapunov equation previously solved is with $n = 110$ in [14].

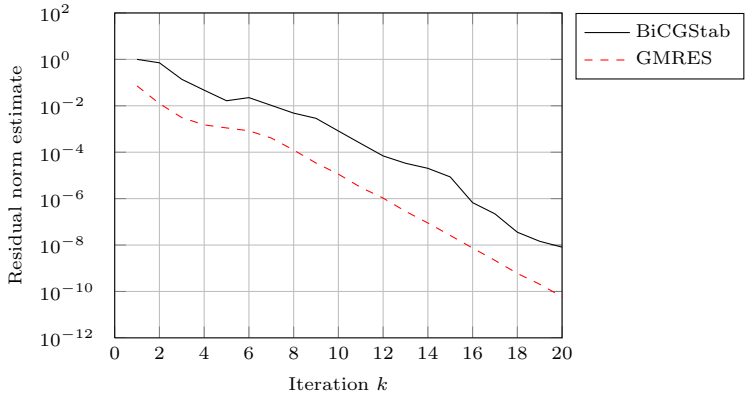


Figure 4: Convergence of the iterative methods with T-Sylvester preconditioning corresponding to the time-delay system stemming from the discretization of the PDDE (22) with $n = 2n_x n_y = 1058$ for the example in Section 4.2.

	Matrix exp. [10]	Discr. first	Proposed method	
	Wall time	Wall time	Wall time	iterations
$n = 28$	0.5 s	0.06 s	3.9 s	19
$n = 50$	296.4 s	0.6 s	12.2 s	21
$n = 242$	MEMERR	30.9 s	232.8 s	25
$n = 722$	MEMERR	MEMERR	4629.3 s	27
$n = 1058$	MEMERR	MEMERR	3.1 hours	27

Table 1: Performance in relation to other methods. MEMERR represents runs which could not solve the problem to reasonable accuracy. Our approach was run with GMRES, RK4 (with $N = 500$) and termination tolerance 10^{-12}. Discr. first represents the approach discussed in [16] and used in [14] with $N = 10$ grid points, which resulted in a solution of accuracy in $U(0)$ in general larger than 10^{-7}.

5. Concluding remarks and outlook

We have in this paper proposed a procedure to solve delay Lyapunov equations based on iterative methods for linear systems combined with a direct method for T-Sylvester equations. Although the method performs well in practice, there appears to be possibilities to improve it further, which we consider beyond the scope of the paper.

As observed in the simulations, the dominating ingredient of the approach is the solution to the T-Sylvester equation. Hence, in order to solve even larger problems we need new methods for T-Sylvester equations. Improvements are possible, e.g., by lower level implementations, or by developing methods which can take the sparsity of the matrices into account, e.g., similar to the Krylov methods and rational Krylov methods for Lyapunov equations [20] or approaches based on Riemannian optimization [32].

The preconditioner in general plays an important role in iterative methods for linear systems and the effectiveness of the preconditioner is typically very problem-dependent. This is also the case in our approach. Although the simulations often worked well, during some experiments, in particular situations where A_0 have some eigenvalues which are very negative, the preconditioner did not appear very effective, even if $\|A_1\|$ was quite small. This can be due to the fact that the hidden constant in the expression (15) may be large.

Since the action of L_c is based on solving an initial value problem, there are very natural options to carry out action of L_c in an inexact way, with a predefined tolerance, thereby saving computation cost. There exists theory for inexact Krylov methods, in e.g., [28], which could be applied in such an approach. However, in our setting, where computing the action of the preconditioner is a dominating cost, an approach with inexact matrix vector product would require more iterations, i.e., more preconditioned actions, and does not appear advantageous in this setting.

The delay Lyapunov equation has been generalized in several ways, e.g., to multiple delays and neutral systems. Our approach might be generalizable to some of these cases. The simplest situations appears to be if the delays are integer multiplies of each other, also known as commensurate delays. For the commensurate case there are procedures which resemble our reformulation (5) with Sylvester resultant matrices [10, Problem 6.72]. However, this increases the size of the problem. An attractive feature of our approach is that we work only with matrices of size n, which would not be the case in the direct adaption to multiple commensurate delays using [10, Problem 6.72].

Acknowledgments

The authors thank Antti Koskela and Tobias Damm for discussions about early results of the paper.

F. Poloni acknowledges the support of the PRA 2014 project "Mathematical models and computational methods for complex networks" of the University of Pisa, and of INDAM (Istituto Nazionale di Alta Matematica). E. Jarlebring

acknowledges the support of the Swedish research council (Vetenskapsrådet) project 2013-4640.

References

References

[1] W. Michiels, S.-I. Niculescu, Stability and Stabilization of Time-Delay Systems: An Eigenvalue-Based Approach, Advances in Design and Control 12, SIAM Publications, Philadelphia, 2007.

[2] K. Gu, V. Kharitonov, J. Chen, Stability of Time-Delay Systems, Control Engineering. Boston, MA: Birkhäuser, 2003.

[3] J.-P. Richard, Time-delay systems: an overview of some recent advances and open problems, Automatica 39 (10) (2003) 1667–1694.

[4] V. Kharitonov, Lyapunov matrices for a class of time delay systems, Syst. Control Lett. 55 (7) (2006) 610–617.

[5] G. Ochoa, V. Kharitonov, S. Mondié, Critical frequencies and parameters for linear delay systems: A Lyapunov matrix approach, Syst. Control Lett. 26 (2013) 781–790.

[6] G. Ochoa, V. Kharitonov, Lyapunov matrices for neutral type time delay systems, in: Proceedings of the 2nd International Conference on Electrical and Electronics Engineering, Mexico City, Mexico, 2005.

[7] G. Ochoa, J. Velázquez-Velázquez, V. Kharitonov, S. Mondié, Lyapunov matrices for neutral type time delay systems, in: Proceedings of the 7th IFAC workshop on time delay systems, Nantes, France, 2007.

[8] S. Mondié, G. Ochoa, B. Ochoa, Instability conditions for linear time delay systems: a Lyapunov matrix function approach, Int. J. Control 84 (10) (2011) 1601–1611.

[9] A. Egorov, S. Mondié, Necessary stability conditions for linear delay systems, Automatica 50 (12) (2014) 3204–3208.

[10] E. Plischke, Transient effects of linear dynamical systems, Ph.D. thesis, Universität Bremen (2005).

[11] V. Kharitonov, D. Hinrichsen, Exponential estimates for time delay systems, Syst. Control Lett. 53 (5) (2004) 395–405.

[12] V. Kharitonov, E. Plischke, Lyapunov matrices for time-delay systems, Syst. Control Lett. 55 (9) (2006) 697–706.

[13] E. Jarlebring, J. Vanbiervliet, W. Michiels, Characterizing and computing the $\mathcal{H}_2$ norm of time-delay systems by solving the delay Lyapunov equation, IEEE Trans. Autom. Control 56 (4) (2011) 814–825.

[14] E. Jarlebring, T. Damm, W. Michiels, Model reduction of time-delay systems using position balancing and delay Lyapunov equations, Math. Control Signals Syst. 25 (2) (2013) 147–166.

[15] E. Huesca, S. Mondié, J. Santos, Polynomial approximations of the Lyapunov matrix of a class of time delay systems, in: Proceedings of the 8th IFAC workshop on time-delay systems, Sinaia, Romania, 2009.

[16] J. Vanbiervliet, W. Michiels, E. Jarlebring, Using spectral discretisation for the optimal $\mathcal{H}_2$ design of time-delay systems, Int. J. Control 84 (2) (2011) 228–241.

[17] A. Merz, Computation of generalized gramians for model reduction of bilinear control systems and time-delay systems, Ph.D. thesis, T.U. Kaiserslautern (2012).

[18] R. Bartels, G. W. Stewart, Solution of the matrix equation $AX + XB = C$, Comm A.C.M. 15 (9) (1972) 820–826.

[19] P. Benner, J.-R. Li, T. Penzl, Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems, Numer. Linear Algebra Appl. 15 (9) (2008) 755–777. doi:10.1002/nla.622.
URL http://dx.doi.org/10.1002/nla.622

[20] V. Simoncini, A new iterative method for solving large-scale Lyapunov matrix equations, SIAM J. Sci. Comput. 29 (3) (2007) 1268–1288.

[21] D. Hu, L. Reichel, Krylov-subspace methods for the Sylvester equation, Linear Algebra Appl. 172 (1992) 283–313.

[22] I. M. Jaimoukha, E. M. Kasenally, Krylov subspace methods for solving large Lyapunov equations, SIAM J. Numer. Anal. 31 (1) (1994) 227–251.

[23] M. Hochbruck, G. Starke, Preconditioned Krylov subspace methods for Lyapunov matrix equations, SIAM J. Matrix Anal. Appl. 16 (1) (1995) 156–171.

[24] V. Simoncini, The Lyapunov matrix equation. Matrix analysis from a computational perspective, Tech. Rep. arXiv:1501.07564, arXiv.org (2015).
URL http://arxiv.org/abs/1501.07564

[25] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.

[26] H. V. der Vorst, BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. and Stat. Comput. 13 (2) (1992) 631–644.

[27] F. De Terán, F. M. Dopico, Consistency and efficient solution of the Sylvester equation for $\star$-congruence, Electron. J. Linear Algebra 22 (2011) 849–863.

[28] V. Simoncini, D. B. Szyld, Theory of inexact Krylov subspace methods and applications to scientific computing, SIAM J. Sci. Comput. 25 (2) (2003) 454–477.

[29] D. Kressner, C. Schröder, D. S. Watkins, Implicit QR algorithms for palindromic and even eigenvalue problems, Numer. Algorithms 51 (2) (2009) 209–238.

[30] Y. Saad, Iterative methods for sparse linear systems, SIAM, 1996.

[31] J. Wu, Theory and Applications of Partial Functional Differential Equations, Applied Mathematical Sciences. 119. New York, NY: Springer., 1996.

[32] B. Vandereycken, S. Vandewalle, A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations, SIAM J. Matrix Anal. Appl. 31 (5) (2010) 2553–2579.

[33] L. Hogben (Ed.), Handbook of linear algebra, 2nd Edition, Discrete Mathematics and its Applications (Boca Raton), CRC Press, Boca Raton, FL, 2014.

Appendix A. Technical bounds

The following results are needed in the proof of Theorem 7.

Lemma 8. *Suppose Z_1 and Z_2 satisfy* (5) *with initial condition $Z_1(0) = Z_2(0) = X$. For $i = 1, 2$,*

$$\|Z_i(t)\|_F \leq 2\exp(2t(\|A_0\| + \|A_1\|))\|X\|_F.$$

Proof. We rely on the vectorized form (6) of the ODE defining $Z_i(t)$; we have

$$\|Z_i(t)\|_F \leq \left\| \begin{bmatrix} \operatorname{vec} Z_1(t) \\ \operatorname{vec} Z_2(t)^T \end{bmatrix} \right\| \leq \|\exp(t\mathcal{A})\| \left\| \begin{bmatrix} \operatorname{vec} X \\ \operatorname{vec} X^T \end{bmatrix} \right\| \leq 2\exp(t\|\mathcal{A}\|)\|X\|_F.$$

To complete the proof, we have to estimate the norm of the matrix $\mathcal{A}$ in (7): we have

$$\|\mathcal{A}\| \leq \|A_0^T \otimes I_n\| + \|A_1^T \otimes I_n\| + \|I_n \otimes A_1^T\| + \|I_n \otimes A_0^T\| = 2(\|A_0\| + \|A_1\|),$$

where we have used the fact that $\|M \otimes N\| = \|M\|\|N\|$. □

Lemma 9. *Suppose that A_0 has no Hamiltonian eigenpairing. Then, there exists a constant K depending only on A_0 and c such that*

$$\|\tilde{L}_c^{-1}(Z)\|_F \leq K\exp(\tau\|A_0\|/2)\|Z\|_F.$$

Proof. Under the stated hypotheses, T is invertible. Let K be the operator norm of T^{-1}, i.e., the smallest constant such that $\|T^{-1}(Z)\|_F \leq K\|Z\|_F$. Then

$$\text{(A.1)} \quad \|\tilde{L}_c^{-1}(Z)\|_F = \|T^{-1}(Z)\exp(\tau A_0/2)\|_F \leq \\ \|T^{-1}(Z)\|_F\|\exp(\tau A_0/2)\| \leq K\|Z\|_F \exp(\tau\|A_0\|/2),$$

where we have used the mixed matrix norm inequality $\|MN\|_F \leq \|M\|_F\|N\|$ [33, Page 50-5, Fact 10]. □