Journal of Computational and Applied Mathematics 384 (2021) 113162

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Modified Douglas splitting method for differential matrix equations

Hao Chen *, Ying Wang

College of Mathematics Science, Chongqing Normal University, Chongqing 401331, China

ARTICLE INFO

Article history:
Received 30 October 2019
Received in revised form 8 August 2020

Keywords:
Modified Douglas splitting method
Differential matrix equation
Differential Lyapunov equation
Differential Riccati equation
Low-rank approximation

ABSTRACT

In this paper, we consider a modified Douglas splitting method for a class of differential matrix equations, including differential Lyapunov and differential Riccati equations. The method we consider is based on a natural three-term splitting of the equations. The implementation of the algorithm requires only the solution of a linear algebraic system with multiple right-hand sides in each time step. It is proved that the method is convergent of order two and it preserves the symmetry and positive semidefiniteness of solutions of differential Lyapunov equations. Moreover, we show how the method can be handled in a low-rank setting for large-scale computations. We also provide a theoretical a priori error analysis for the low-rank algorithms. Numerical results are presented to verify the theoretical analysis.

1. Introduction

We are interested in the numerical solution of the following differential matrix equation (DME)

$$\dot{X}(t) = AX(t) + X(t)A^T + G(t, X(t)), \qquad X(t_0) = X_0, \tag{1.1}$$

where $X(t), A \in \mathbb{R}^{d\times d}$, $G : [t_0, T] \times \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$, and $(\cdot)^T$ denotes the transpose. The most common DMEs are the well-known differential Riccati and differential Lyapunov equations (DREs/DLEs), where the DLEs can be considered as a special case of the DREs. DREs and DLEs have been proposed and investigated in many areas, such as optimal control problems and model reduction of linear dynamical systems (see [1–5] and the references therein).

A typical application of the DREs is a linear quadratic regulator problem (LQR). Consider the linear control system

$$\dot{x} = Ax + Bu, \quad x(t_0) = x_0, \tag{1.2}$$

$$y = Cx, \tag{1.3}$$

where $A \in \mathbb{R}^{d\times d}$, $B \in \mathbb{R}^{d\times m}$, $C \in \mathbb{R}^{r\times d}$ are given matrices, $x \in \mathbb{R}^d$ is the state and $u \in \mathbb{R}^m$ is the control. The goal is to drive the output y to a given target. The optimal control function u^* is found by minimizing the cost functional

$$J(u) = \int_{t_0}^{T} \left(x(t)^T C^T Q C x(t) + u(t)^T R u(t)\right) dt, \tag{1.4}$$

* Corresponding author.
E-mail address: hch@cqnu.edu.cn (H. Chen).

https://doi.org/10.1016/j.cam.2020.113162
0377-0427/

where $Q \in \mathbb{R}^{r\times r}$, $R \in \mathbb{R}^{m\times m}$ are given weighting matrices. It can be shown (see [1] for details) that u^* is given in feedback form as $u^*(t) = -R^{-1}B^TX(T-t)x(t)$, where $X(t)$ is the solution of the following DRE

$$\dot{X}(t) = A^TX(t) + X(t)A + C^TQC - X(t)BR^{-1}B^TX(t). \tag{1.5}$$

We remark that if the solution of the DRE (1.5) converges for $T \to \infty$ to a steady state, this steady state is then given as the solution of the algebraic Riccati equation (ARE)

$$0 = A^TX + XA + C^TQC - XBR^{-1}B^TX. \tag{1.6}$$

The development of efficient numerical methods for solving the DMEs is a very active topic. Besides matrix versions of classical implicit time discretization schemes such as Rosenbrock methods, midpoint and trapezoidal rules, and BDF [6–9] for stiff DMEs we mention, among others, peer methods [10], splitting methods [11–16] and Krylov projection methods [17]. It has been shown that the solution to the DMEs that arise in practical applications often is of low numerical rank, see [18–22]. Therefore, we can factorize $X = ZZ^T$ where Z is a matrix with many fewer columns than X. The idea of low-rank algorithms is then to do computations on the factor Z instead of X. We remark that a decomposition of the form $X = ZZ^T$ is just one possibility. Other types of low-rank factorization do exist. A number of low-rank based solution algorithms for large-scale DMEs have been proposed and studied, see for instance [6,7,9,10,12–15,23–25]. We remark that for the solution of the DREs, most of the above mentioned methods require to solve a number of either AREs or algebraic Lyapunov equations (ALEs) in each time step. We refer the readers to [26,27] and the references therein for discussion on efficient solvers for AREs and ALEs.

Here we are interested in the numerical solution of the DMEs (1.1) by applying a modified Douglas splitting (MDS) scheme [28]. The Douglas methods, also referred to as alternating direction implicit (ADI) methods, were originally designed for solving multi-dimensional parabolic problems, see [29,30] and also [31]. By adding a forward Euler step for the explicit term in the discretization, a Douglas stabilizing correction method has been introduced in [32], but the order of convergence is only one. The MDS method was first introduced in [28] to solve multi-dimensional reaction–diffusion equations, where explicit terms are treated as in the explicit trapezoidal rule and the method is second-order convergent. It has been shown in [28] that the MDS scheme is competitive for solving multi-dimensional reaction–diffusion equations.

In this paper, a matrix-valued version of the MDS scheme is considered for solving the DMEs (1.1). The method is based on a natural three-term splitting of the equations. Its implementation does not require to solve any AREs or ALEs. Generally, the implementation of the MDS scheme only requires to solve a linear system with multiple right-hand sides in each time step. We demonstrate that the MDS scheme is second-order convergent and it preserves symmetry and positive semidefiniteness of the DLEs. We remark that the convergence analysis of the MDS scheme for the DMEs stated in this paper is different from the analysis presented in [28], where the convergence proof is only performed for linear problems. Furthermore, low-rank MDS methods have been developed for the DLEs and DREs, respectively. We also present a theoretical a priori error analysis for the proposed low-rank algorithms.

The rest of this paper is organized as follows. In Section 2, we discuss the MDS scheme for the DMEs (1.1) and prove its convergence. We also study some structure-preserving properties of the MDS scheme for DLEs. In Section 3, we develop and analyze low-rank versions of the MDS method for solving DLEs and DREs. In Section 4, we present numerical results to verify the theoretical results. Finally, we give concluding remarks in Section 5.

2. Modified Douglas splitting method

2.1. Modified Douglas splitting method

The aim of this section is to discuss the so-called MDS scheme for the discretization of the DMEs (1.1). The MDS scheme can be obtained as follows, see [28] for details. Consider the following initial value problem

$$x'(t) = f(t, x(t)), \quad x(t_0) = x_0. \tag{2.1}$$

Here we assume that $f(t,x)$ has the following decomposition

$$f(t,x) = f_0(t,x) + f_1(t,x) + \cdots + f_s(t,x) \tag{2.2}$$

in which the separate component function f_j are more simple than the whole f. Usually f_0 is a nonstiff nonlinear term that can be treated explicitly in a time integration scheme. The other f_j will be treated in an implicit fashion. Consider the time interval $[t_0, T]$, and approximate it with a uniform mesh of size τ and $N+1$ points, such that $t_0 < t_1 < \cdots < t_N = T$ and $t_n = t_0 + n\tau$, $n = 0, \ldots, N$. Denoting by x_n the discrete approximation to $x(t_n)$, the MDS method for (2.1) is given by

$$\begin{cases} \tilde{x}_{n+1} = x_n + \tau f(t_n, x_n), \\ z_0 = \tilde{x}_{n+1} + \dfrac{1}{2}\tau\left(f_0(t_{n+1}, \tilde{x}_{n+1}) - f_0(t_n, x_n)\right), \\ z_j = z_{j-1} + \dfrac{1}{2}\tau\left(f_j(t_{n+1}, z_j) - f_j(t_n, x_n)\right), \quad j = 1, 2, \ldots, s, \\ x_{n+1} = z_s. \end{cases} \tag{2.3}$$

Now, we apply the MDS scheme for the approximation of the DMEs (1.1). Denoting by X_n the discrete approximation to $X(t_n)$ for $n = 0, \ldots, N$. Let $f_0 = G(t, X), f_1 = AX, f_2 = XA^T$, then the MDS discretization of the problem (1.1) is as follows

$$\begin{cases} \tilde{X}_{n+1} = X_n + \tau(AX_n + X_nA^T + G(t_n, X_n)), \\ Z_0 = \tilde{X}_{n+1} + \dfrac{1}{2}\tau\left(G(t_{n+1}, \tilde{X}_{n+1}) - G(t_n, X_n)\right), \\ Z_1 = Z_0 + \dfrac{1}{2}\tau(AZ_1 - AX_n), \\ Z_2 = Z_1 + \dfrac{1}{2}\tau(Z_2A^T - X_nA^T), \\ X_{n+1} = Z_2. \end{cases} \tag{2.4}$$

A sketch of the MDS procedure for the solution of DMEs (1.1) is given in *Algorithm* 1. Note that in each time step of the MDS scheme, we only need to solve a linear algebraic system with the coefficient matrix $I - \frac{\tau}{2}A$ and multiple right-hand sides. We can solve the linear system associated with $I - \frac{\tau}{2}A$ by its LU factorization.

Algorithm 1 MDS scheme for DMEs (1.1)

1: Set $\tau = (T - t_0)/N$ and $t_n = t_0 + \tau n$;
2: LU factorize $I - \frac{1}{2}\tau A := LU$;
3: **for** $n = 0$ **to** $N - 1$ **do**
4: Set $\tilde{X}_{n+1} = X_n + \tau(AX_n + X_nA^T + G(t_n, X_n))$;
5: Set $Z_0 = \tilde{X}_{n+1} + \frac{1}{2}\tau(G(t_{n+1}, \tilde{X}_{n+1}) - G(t_n, X_n))$;
6: Solve $LUZ_1 = Z_0 - \frac{\tau}{2}AX_n$;
7: Solve $LUZ_2^T = (Z_1 - \frac{\tau}{2}X_nA^T)^T$;
8: Set $X_{n+1} = Z_2$;
9: **end for**

2.2. Convergence analysis

In this section, we focus on the convergence analysis of the MDS scheme (2.4). For proving convergence, we need the following assumption.

Assumption 1. We assume that the following properties hold.

- There exists $\omega > 0$ such that the matrix A satisfies
$$\left\| \left(I - \frac{1}{2}\tau A\right)^{-1} \left(I + \frac{1}{2}\tau A\right) \right\|_2 \leq 1, \quad \left\| \left(I - \frac{1}{2}\tau A\right)^{-1} \right\|_2 \leq 1, \quad \|I \otimes A + A \otimes I\|_2 \leq \omega$$
for all $\tau > 0$.
- Both $G(t, X)$ and the exact solution of the DME (1.1) are sufficiently smooth. As a result, there exists $L > 0$ such that
$$\|G(t, Y) - G(t, Z)\| \leq L\|Y - Z\|,$$
for $t_0 \leq t \leq T$ and all $Y, Z \in \mathbb{R}^{d\times d}$.

Here and in the rest of this paper, $\|\cdot\|$ denotes the Frobenius norm. We remark that DMEs of the form (1.1) typically arise from parabolic partial differential equations. In this paper we focus on DMEs that stem from problems associated with diffusion equations. In this case, the matrix A is the discretization of the diffusion operator and $-A$ is typically symmetric and positive definite (or semidefinite). It then follows that the first property of Assumption 1 holds. We remark that a similar assumption is given in [14] to prove the convergence of the Lie–Trotter splitting method for DMEs.

Before presenting the convergence results, a brief review of some properties related to Kronecker products is needed, see [33] for details. Let $C, B, D, E \in \mathbb{R}^{d\times d}$, then

$$(C \otimes B)^T = C^T \otimes B^T, \quad (C \otimes B)(D \otimes E) = (CD) \otimes (BE),$$
$$\sigma(C \otimes B) = \{\lambda_i\mu_j : \ \lambda_i \in \sigma(C) \text{ and } \mu_j \in \sigma(B), i, j = 1, \ldots, d\},$$
$$(C \otimes B)z = \mathrm{vec}(BZC^T), \ \mathrm{vec}(Z) = z,$$

where $\sigma(\cdot)$ denotes the spectrum of a matrix and the vec operator transforms matrices into vectors by stacking columns as follows

$$Z = [z_1, z_2, \ldots, z_d] \in \mathbb{R}^{d\times d} \iff \mathrm{vec}(Z) = [z_1^T, z_2^T, \ldots, z_d^T]^T \in \mathbb{R}^{d^2}.$$

Note that $\|Z\| = \|\mathrm{vec}(Z)\|_2 = \|z\|_2$. It then follows that

$$\|BZC^T\| = \|(C \otimes B)z\|_2 \le \|C \otimes B\|_2 \|z\|_2 = \|C \otimes B\|_2 \|Z\|. \tag{2.5}$$

It is easy to show that

$$\begin{aligned}\|C \otimes B\|_2 &= \sqrt{\rho((C \otimes B)^T (C \otimes B))} = \sqrt{\rho((C^T C) \otimes (B^T B))} \\ &= \sqrt{\rho(C^T C)\rho(B^T B)} = \|C\|_2 \|B\|_2,\end{aligned}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix. Then from (2.5) we have

$$\|BZC^T\| \le \|C\|_2 \|B\|_2 \|Z\|. \tag{2.6}$$

The following results will be useful in the proof of our convergence results.

Lemma 2.1. *Assume that the first property of Assumption 1 holds, then we have*

$$\left\| \left(I - \frac{1}{2}\tau A\right)^{-1} \left(I + \frac{1}{2}\tau A\right) Z \left(I + \frac{1}{2}\tau A^T\right) \left(I - \frac{1}{2}\tau A^T\right)^{-1} \right\| \le \|Z\|, \tag{2.7}$$

$$\left\| \left(I - \frac{1}{2}\tau A\right)^{-1} Z \left(I - \frac{1}{2}\tau A^T\right)^{-1} \right\| \le \|Z\|, \tag{2.8}$$

$$\|AZ + ZA^T\| \le \omega \|Z\| \tag{2.9}$$

for all $\tau > 0$ and all $Z \in \mathbb{R}^{d\times d}$.

Proof. The inequalities (2.7)–(2.8) follow directly from (2.6) and Assumption 1. Denote $\mathrm{vec}(Z) = z$, we have

$$\begin{aligned}\|AZ + ZA^T\| &= \|\mathrm{vec}(AZ + ZA^T)\|_2 = \|(I \otimes A + A \otimes I)z\|_2 \\ &\le \|I \otimes A + A \otimes I\|_2 \|z\|_2 \le \omega \|Z\|,\end{aligned}$$

where we have used the assumption $\|I \otimes A + A \otimes I\|_2 \le \omega$. □

Now we are in the position to show the convergence of the MDS scheme.

Theorem 2.2. *Assume that Assumption 1 holds, then the MDS scheme (2.4) is second-order convergent.*

Proof. Using the formula for the variation of constants, the solution of the DME (1.1) at time $t_{n+1} = t_n + \tau$ can be written as

$$X(t_{n+1}) = e^{\tau A} X(t_n) e^{\tau A^T} + \int_0^\tau e^{(\tau - s)A} G(t_n + s, X(t_n + s)) e^{(\tau - s)A^T} ds. \tag{2.10}$$

By applying the midpoint rule $\int_0^\tau g(s)ds = \tau g\left(\frac{\tau}{2}\right) + O(\tau^3)$ to (2.10), we have that

$$X(t_{n+1}) = e^{\tau A} X(t_n) e^{\tau A^T} + \tau e^{\frac{\tau A}{2}} G\left(t_n + \frac{\tau}{2}, X\left(t_n + \frac{\tau}{2}\right)\right) e^{\frac{\tau A^T}{2}} + O(\tau^3).$$

Then applying the midpoint rule $g\left(\frac{\tau}{2}\right) = \frac{1}{2}[g(0) + g(\tau)] + O(\tau^2)$ to G in the above equation, we get

$$\begin{aligned}X(t_{n+1}) &= e^{\tau A} X(t_n) e^{\tau A^T} + \frac{\tau}{2} e^{\frac{\tau A}{2}} \left[G(t_n, X(t_n)) + G(t_{n+1}, X(t_{n+1})) + O(\tau^2)\right] e^{\frac{\tau A^T}{2}} + O(\tau^3) \\ &= e^{\tau A} X(t_n) e^{\tau A^T} + \frac{\tau}{2} e^{\frac{\tau A}{2}} \left[G(t_n, X(t_n)) + G(t_{n+1}, X(t_{n+1}))\right] e^{\frac{\tau A^T}{2}} + O(\tau^3).\end{aligned}$$

Now, consider the following Taylor expansions

$$\begin{aligned}&e^{\tau A} = I + \tau A + \frac{\tau^2}{2}A^2 + \frac{\tau^3}{6}A^3 + O(\tau^4), \quad e^{\frac{\tau A}{2}} = I + \frac{\tau}{2}A + \frac{\tau^2}{8}A^2 + O(\tau^3), \\ &\left(I - \frac{1}{2}\tau A\right)^{-1} = I + \frac{\tau}{2}A + \frac{\tau^2}{4}A^2 + O(\tau^3), \\ &\left(I - \frac{1}{2}\tau A\right)^{-1} \left(I + \frac{1}{2}\tau A\right) = I + \tau A + \frac{\tau^2}{2}A^2 + \frac{\tau^3}{4}A^3 + O(\tau^4),\end{aligned}$$

we have that

$$e^{\tau A} = \left(I - \frac{1}{2}\tau A\right)^{-1} \left(I + \frac{1}{2}\tau A\right) + O(\tau^3), \quad e^{\frac{\tau A}{2}} = \left(I - \frac{1}{2}\tau A\right)^{-1} + O(\tau^2).$$

It then follows that

$$X(t_{n+1}) = \left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)X(t_n)\left(I + \frac{1}{2}\tau A^T\right)\left(I - \frac{1}{2}\tau A^T\right)^{-1} + \frac{\tau}{2}\left(I - \frac{1}{2}\tau A\right)^{-1}[G(t_n, X(t_n)) + G(t_{n+1}, X(t_{n+1}))]\left(I - \frac{1}{2}\tau A^T\right)^{-1} + O(\tau^3). \tag{2.11}$$

Consider the Taylor expansions

$$X(t_{n+1}) = X(t_n) + \tau\dot{X}(t_n) + O(\tau^2) := \tilde{X}(t_{n+1}) + O(\tau^2),$$

and

$$G(t_{n+1}, X(t_{n+1})) = G(t_{n+1}, \tilde{X}(t_{n+1}) + O(\tau^2)) = G(t_{n+1}, \tilde{X}(t_{n+1})) + O(\tau^2).$$

By using the above Taylor expansion of $G(t_{n+1}, X(t_{n+1}))$ and the formula (2.11), we have that

$$X(t_{n+1}) = \left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)X(t_n)\left(I + \frac{1}{2}\tau A^T\right)\left(I - \frac{1}{2}\tau A^T\right)^{-1} + \frac{\tau}{2}\left(I - \frac{1}{2}\tau A\right)^{-1}\left[G(t_n, X(t_n)) + G(t_{n+1}, \tilde{X}(t_{n+1}))\right]\left(I - \frac{1}{2}\tau A^T\right)^{-1} + O(\tau^3). \tag{2.12}$$

By eliminating the intermediate matrices Z_0, Z_1, Z_2 in (2.4), we obtain

$$X_{n+1} = \left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)X_n\left(I + \frac{1}{2}\tau A^T\right)\left(I - \frac{1}{2}\tau A^T\right)^{-1} + \frac{\tau}{2}\left(I - \frac{1}{2}\tau A\right)^{-1}\left[G(t_n, X_n) + G(t_{n+1}, \tilde{X}_{n+1})\right]\left(I - \frac{1}{2}\tau A^T\right)^{-1}. \tag{2.13}$$

Subtracting (2.13) from (2.12) and defining the local error $E_n := X(t_n) - X_n$, we obtain the following

$$E_{n+1} = \left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)E_n\left(I + \frac{1}{2}\tau A^T\right)\left(I - \frac{1}{2}\tau A^T\right)^{-1} + \frac{\tau}{2}\left(I - \frac{1}{2}\tau A\right)^{-1}[G(t_n, X(t_n)) - G(t_n, X_n)]\left(I - \frac{1}{2}\tau A^T\right)^{-1} + \frac{\tau}{2}\left(I - \frac{1}{2}\tau A\right)^{-1}\left[G(t_{n+1}, \tilde{X}(t_{n+1})) - G(t_{n+1}, \tilde{X}_{n+1})\right]\left(I - \frac{1}{2}\tau A^T\right)^{-1} + O(\tau^3). \tag{2.14}$$

Taking norms and using Lemma 2.1, Eq. (2.14) leads to

$$\begin{aligned}\|E_{n+1}\| &\le \|E_n\| + \frac{\tau}{2}\|G(t_n, X(t_n)) - G(t_n, X_n)\| \\ &\quad + \frac{\tau}{2}\|G(t_{n+1}, \tilde{X}(t_{n+1})) - G(t_{n+1}, \tilde{X}_{n+1})\| + c_1\tau^3 \\ &\le \|E_n\| + \frac{\tau L}{2}(\|E_n\| + \|\tilde{X}(t_{n+1}) - \tilde{X}_{n+1}\|) + c_1\tau^3,\end{aligned} \tag{2.15}$$

where c_1 is a positive constant and we have used the Lipschitz condition as stated in Assumption 1. Note that

$$\begin{aligned}\|\tilde{X}(t_{n+1}) - \tilde{X}_{n+1}\| &= \|X(t_n) + \tau\dot{X}(t_n) - X_n - \tau(AX_n + X_nA^T + G(t_n, X_n))\| \\ &\le \|E_n\| + \tau\|AE_n + E_nA^T\| + \tau\|G(t_n, X(t_n)) - G(t_n, X_n)\| \\ &\le (1 + \tau\omega + \tau L)\|E_n\|,\end{aligned}$$

where we have used (2.9) and the Lipschitz condition on G. Now, we define $e_n := \|E_n\|$ and write (2.15) as

$$e_{n+1} \le \left(1 + \tau L + \frac{\tau^2 L(\omega + L)}{2}\right)e_n + c_1\tau^3 := (1 + \tau\alpha)e_n + c_1\tau^3, \tag{2.16}$$

where $\alpha = L + \tau L(\omega + L)/2$. It follows by induction that

$$e_n \le (1 + \tau\alpha)^n e_0 + \left((1 + \tau\alpha)^n - 1\right)\frac{c_1\tau^2}{L + \tau L(\omega + L)/2}.$$

Now, notice that $e_0 = \|E_0\| = \|X(t_0) - X_0\| = 0$ and $1 + \tau\alpha \le e^{\tau\alpha}$, and recalling that $e_n = \|X(t_n) - X_n\|$, we obtain that

$$\begin{aligned}\|X(t_n) - X_n\| &\le \left(e^{n\tau\alpha} - 1\right) \frac{2c_1\tau^2}{2L + \tau L(\omega + L)} \\ &\le \left(e^{(T-t_0)(L+\tau L(\omega+L)/2)} - 1\right) \frac{2c_1\tau^2}{2L + \tau L(\omega + L)}.\end{aligned}$$

The previous results shows that the MDS scheme (2.4) is convergent and second-order accurate. □

Note that Eq. (1.1) can represent the DLE

$$\dot{X}(t) = AX(t) + X(t)A^T + Q(t), \qquad X(t_0) = X_0. \tag{2.17}$$

In this case, $G(t, X) = Q(t)$ with $Q(t)$ being a matrix function. Let $Q(t)$ and the initial value X_0 be symmetric and positive semidefinite, then the solution of the DLE (2.17) is also symmetric and positive semidefinite, see [1]. The following result shows that the MDS scheme preserves symmetry and positive semidefiniteness of the solution of the DLEs.

Theorem 2.3. *The MDS scheme (2.4) preserves the symmetry and positive semidefiniteness of the solution of the DLE (2.17).*

Proof. Replacing $G(t, X) = Q(t)$ in (2.13), we obtain

$$\begin{aligned}X_{n+1} = &\left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)X_n\left(I + \frac{1}{2}\tau A^T\right)\left(I - \frac{1}{2}\tau A^T\right)^{-1} \\ &+ \left(I - \frac{1}{2}\tau A\right)^{-1}\left[\frac{\tau}{2}Q(t_n) + \frac{\tau}{2}Q(t_{n+1})\right]\left(I - \frac{1}{2}\tau A^T\right)^{-1}.\end{aligned} \tag{2.18}$$

Then by the symmetry and positive semidefiniteness of X_0 and $Q(t_n)$ $(n = 0, \ldots, N)$, we can conclude that X_n $(n = 1, \ldots, N)$ are symmetric and positive semidefinite. □

3. Low-rank MDS methods for DLEs and DREs

For large-scale DLEs and DREs, it is necessary to avoid forming the solution matrices X_n explicitly, because this in general leads to dense computations. In practical applications the input data are usually given in a low-rank representation. In these situations it is observed that the solution is also of low numerical rank, see [18–22] for details. This indicates that one can use low-rank representation based algorithms to solve large-scale DLEs and DREs. The development of efficient low-rank algorithms for solving large-scale DLEs and DREs is a very active topic; see, e.g., [6,7,9,10,12–15,23–26]. In this section, we will develop low-rank versions of the MDS method for DLEs and DREs.

3.1. Low-rank MDS method for DLEs

Now assume that $Q(t_n)$ are given in the form $Q(t_n) = C_n^T C_n$ for $n = 0, \ldots, N$ with $C_n \in \mathbb{R}^{q\times d}$, $q \ll d$. And assume that X_n admits a decomposition of the form $X_n = Z_n Z_n^T$ for $n = 0, \ldots, N$ with $Z_n \in R^{d\times k}$. The approximation X_{n+1} can also be computed in factorized form $X_{n+1} = Z_{n+1}Z_{n+1}^T$ with

$$Z_{n+1} = \left(I - \frac{1}{2}\tau A\right)^{-1}\left[\left(I + \frac{1}{2}\tau A\right)Z_n, \sqrt{\frac{\tau}{2}}C_n^T, \sqrt{\frac{\tau}{2}}C_{n+1}^T\right] \in \mathbb{R}^{d\times(k+2q)}. \tag{3.1}$$

This easily follows from (2.18) when plugging in the factorized form of X_n, $Q(t_n)$, and $Q(t_{n+1})$.

Note that the new matrix Z_{n+1} has more columns than either Z_n or C_n^T, and likely also more than its rank. A better low-rank approximation $\hat{Z}_{n+1}$ can be found by applying a column compression technique. Here we apply the classic Rank-Revealing QR (RRQR) based column compression, see [24] for details. In the following we employ MATLAB notation to specify subblocks of a matrix.

RRQR based column compression:

- Compute $Z^T = QRP^T$ with $Z \in \mathbb{R}^{d\times p}$, $Q \in \mathbb{R}^{p\times p}$, $Q^TQ = I_p$, $R \in \mathbb{R}^{p\times d}$ and a permutation matrix $P \in \mathbb{R}^{d\times d}$.
- Set $Z_r = PR_r^T \in \mathbb{R}^{d\times r}$, where $R_r := R(1:r, :) \in \mathbb{R}^{r\times d}$ and $r < p$ is the required rank.

Denote $Q_r = Q(1:r, 1:r) \in \mathbb{R}^{r\times r}$ and set $r = p$, it follows that

$$Z_r Z_r^T = PR_r^T R_r P^T = PR_r^T Q_r^T Q_r R_r P^T = PR^T Q^T QRP^T = ZZ^T.$$

Therefore, we obtain that $Z_r Z_r^T \approx ZZ^T$ for $r < p$.

In Algorithm 2 we summarize the proposed low-rank MDS method for the solution of DLEs (2.17).

Algorithm 2 Low-rank MDS method for DLEs (2.17)

Require: Low-rank factors $\hat{Z}_0, C_n$ such that $X_0 = \hat{Z}_0\hat{Z}_0^T$, $Q(t_n) = C_n^T C_n$.
Ensure: $(\hat{Z}_{n+1}, t_{n+1})$ such that $X_{n+1} \approx \hat{X}_{n+1} := \hat{Z}_{n+1}\hat{Z}_{n+1}^T$.
1: Set $\tau = (T - t_0)/N$ and $t_n = t_0 + \tau n$;
2: LU factorize $I - \frac{1}{2}\tau A := LU$;
3: **for** $n = 0$ **to** $N - 1$ **do**
4: Solve $LU\bar{Z}_{n+1} = \left[\left(I + \frac{1}{2}\tau A\right)\hat{Z}_n, \sqrt{\frac{\tau}{2}}C_n^T, \sqrt{\frac{\tau}{2}}C_{n+1}^T\right]$;
5: Column-compress $\hat{Z}_{n+1} \approx \bar{Z}_{n+1}$ by, e.g., RRQR;
6: **end for**

Theorem 3.1. *Assume that Assumption 1 holds and the column compression error in Algorithm 2 satisfies* $\|\bar{Z}_{n+1} - \hat{Z}_{n+1}\| \le \varepsilon$, $n = 0, 1, \ldots, N-1$. *Then the numerical solution* $\hat{X}_n$ *of the low-rank MDS algorithm for DLEs (2.17) satisfies*

$$\|X(t_n) - \hat{X}_n\| \le c_1(T - t_0)\tau^2 + \frac{c_2(T - t_0)\varepsilon}{\tau}, \quad n = 0, 1, \ldots, N,$$

where c_1, c_2 *are positive constants.*

Proof. From Algorithm 2, we have

$$\bar{Z}_{n+1} = \left(I - \frac{1}{2}\tau A\right)^{-1}\left[\left(I + \frac{1}{2}\tau A\right)\hat{Z}_n, \sqrt{\frac{\tau}{2}}C_n^T, \sqrt{\frac{\tau}{2}}C_{n+1}^T\right].$$

Subtracting the above equation from (3.1) and taking norms we obtain

$$\begin{aligned}
\|Z_{n+1} - \bar{Z}_{n+1}\| &= \left\|\left[\left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)(Z_n - \hat{Z}_n), 0_{d\times q}, 0_{d\times q}\right]\right\| \\
&= \left\|\left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)(Z_n - \hat{Z}_n)\right\| \\
&\le \left\|\left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)\right\|_2 \|Z_n - \hat{Z}_n\| \\
&\le \|Z_n - \hat{Z}_n\|, \qquad (3.2)
\end{aligned}$$

where $0_{d\times q}$ denotes $d \times q$ zero matrix and we have used (2.6) and Assumption 1. In addition, by using the inequality (3.2) and the assumption that $\|\bar{Z}_{n+1} - \hat{Z}_{n+1}\| \le \varepsilon$, we can show that

$$\begin{aligned}
\|Z_{n+1} - \hat{Z}_{n+1}\| &= \|Z_{n+1} - \bar{Z}_{n+1} + \bar{Z}_{n+1} - \hat{Z}_{n+1}\| \\
&\le \|Z_{n+1} - \bar{Z}_{n+1}\| + \|\bar{Z}_{n+1} - \hat{Z}_{n+1}\| \\
&\le \|Z_n - \hat{Z}_n\| + \varepsilon. \qquad (3.3)
\end{aligned}$$

Note that

$$\begin{aligned}
\|X_{n+1} - \hat{X}_{n+1}\| &= \|Z_{n+1}Z_{n+1}^T - \hat{Z}_{n+1}\hat{Z}_{n+1}^T\| \\
&= \|Z_{n+1}Z_{n+1}^T - \hat{Z}_{n+1}Z_{n+1}^T + \hat{Z}_{n+1}Z_{n+1}^T - \hat{Z}_{n+1}\hat{Z}_{n+1}^T\| \\
&\le \|Z_{n+1}^T\|\|Z_{n+1} - \hat{Z}_{n+1}\| + \|\hat{Z}_{n+1}\|\|Z_{n+1}^T - \hat{Z}_{n+1}^T\| \\
&= (\|Z_{n+1}\| + \|\hat{Z}_{n+1}\|)\|Z_{n+1} - \hat{Z}_{n+1}\| \\
&\le c_2(\|Z_n - \hat{Z}_n\| + \varepsilon), \qquad (3.4)
\end{aligned}$$

where we have used the inequality (3.3) and $c_2 := \max_{n=0,1,\ldots,N}(\|Z_n\| + \|\hat{Z}_n\|)$.

From (2.14)–(2.15) and $G(t, X) = Q(t)$ we have

$$\|X(t_{n+1}) - X_{n+1}\| \le \|X_n - X(t_n)\| + c_1\tau^3. \qquad (3.5)$$

It then follows from (3.4) and (3.5) that

$$\begin{aligned}
\|X(t_{n+1}) - \hat{X}_{n+1}\| &= \|X(t_{n+1}) - X_{n+1} + X_{n+1} - \hat{X}_{n+1}\| \\
&\le \|X(t_{n+1}) - X_{n+1}\| + \|X_{n+1} - \hat{X}_{n+1}\| \\
&\le \|X_n - X(t_n)\| + c_1\tau^3 + c_2(\|Z_n - \hat{Z}_n\| + \varepsilon).
\end{aligned}$$

By induction we obtain that

$$\|X(t_n) - \hat{X}_n\| \le \|X(t_0) - X_0\| + c_1 n\tau^3 + c_2(\|Z_0 - \hat{Z}_0\| + n\varepsilon).$$

Then by $\|X(t_0) - X_0\| = 0$, $\|Z_0 - \hat{Z}_0\| = 0$ and $n\tau \le (T - t_0)$ the result follows. □

3.2. Low-rank MDS method for DREs

We remark that DMEs (1.1) can also represent the DREs

$$\dot{X}(t) = AX(t) + X(t)A^T + Q - XPX, \qquad X(t_0) = X_0. \tag{3.6}$$

In this case, $G(t, X) = Q - XPX$ where Q and P are constant matrices. If Q, P and the initial value X_0 are symmetric and positive semidefinite, then the symmetry and positive semidefiniteness of the solution of the DRE (3.6) is guaranteed, see [34].

By replacing $G(t, X) = Q - XPX$ in (2.13), the MDS scheme for the DREs can be written as

$$\begin{aligned} X_{n+1} = &\left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)X_n\left(I + \frac{1}{2}\tau A^T\right)\left(I - \frac{1}{2}\tau A^T\right)^{-1} \\ &+\frac{\tau}{2}\left(I - \frac{1}{2}\tau A\right)^{-1}\left[2Q - X_nPX_n - \tilde{X}_{n+1}P\tilde{X}_{n+1}\right]\left(I - \frac{1}{2}\tau A^T\right)^{-1}, \end{aligned} \tag{3.7}$$

where

$$\tilde{X}_{n+1} = X_n + \tau(AX_n + X_nA^T + Q - X_nPX_n). \tag{3.8}$$

It is easy to show that X_n $(n = 1, \ldots, N)$ are symmetric under the conditions that P, Q, X_0 are symmetric. The determination of the positive semidefiniteness of X_n appears to be a difficult problem. However, numerical tests (see Section 4) show that the MDS scheme can also preserve positive semidefiniteness of the solutions of the DREs.

Now we assume that there are low-rank factorizations C and B of Q and P, respectively, i.e., $Q = C^TC$, $P = BB^T$ with $C \in \mathbb{R}^{q\times d}$, $B \in \mathbb{R}^{d\times m}$, $q, m \ll d$. And assume that X_n admits a decomposition of the form $X_n = Z_nZ_n^T$ with $Z_n \in R^{d\times k}$ for $n = 0, \ldots, N$. It follows from (3.8) that

$$\begin{aligned} \tilde{X}_{n+1} &= \left(\frac{1}{2}I + \tau A\right)X_n + X_n\left(\frac{1}{2}I + \tau A\right)^T + \tau Q - \tau X_nPX_n \\ &= \left(\frac{1}{2}I + \tau A\right)Z_nZ_n^T + Z_nZ_n^T\left(\frac{1}{2}I + \tau A\right)^T + \tau C^TC - \tau Z_nZ_n^TBB^TZ_nZ_n^T \\ &= \tau C^TC + \left[\left(\frac{1}{2}I + \tau A\right)Z_n + Z_n\right]\left[\left(\frac{1}{2}I + \tau A\right)Z_n + Z_n\right]^T \\ &\quad - \tau Z_nZ_n^TBB^TZ_nZ_n^T - \left(\frac{1}{2}I + \tau A\right)Z_nZ_n^T\left(\frac{1}{2}I + \tau A\right)^T - Z_nZ_n^T \\ &= \tilde{Z}_{n+1}\tilde{Z}_{n+1}^T, \end{aligned}$$

where

$$\tilde{Z}_{n+1} = \left[\sqrt{\tau}C^T, \left(\frac{3}{2}I + \tau A\right)Z_n, i\sqrt{\tau}Z_nZ_n^TB, i\left(\frac{1}{2}I + \tau A\right)Z_n, iZ_n\right] \in \mathbb{C}^{d\times(3k+q+m)}. \tag{3.9}$$

Here $i = \sqrt{-1}$. It then follows from (3.7) that

$$\begin{aligned} X_{n+1} = &\left(I - \frac{1}{2}\tau A\right)^{-1}\left(I + \frac{1}{2}\tau A\right)Z_nZ_n^T\left(I + \frac{1}{2}\tau A^T\right)\left(I - \frac{1}{2}\tau A^T\right)^{-1} \\ &+\frac{\tau}{2}\left(I - \frac{1}{2}\tau A\right)^{-1}\left(2C^TC - Z_nZ_n^TBB^TZ_nZ_n^T\right)\left(I - \frac{1}{2}\tau A^T\right)^{-1} \\ &-\frac{\tau}{2}\left(I - \frac{1}{2}\tau A\right)^{-1}\left(\tilde{Z}_{n+1}\tilde{Z}_{n+1}^TBB^T\tilde{Z}_{n+1}\tilde{Z}_{n+1}^T\right)\left(I - \frac{1}{2}\tau A^T\right)^{-1}, \\ = &\, Z_{n+1}Z_{n+1}^T, \end{aligned}$$

where

$$Z_{n+1} = \left(I - \frac{1}{2}\tau A\right)^{-1}\left[\sqrt{\tau}C^T, \left(I + \frac{1}{2}\tau A\right)Z_n, i\sqrt{\frac{\tau}{2}}Z_nZ_n^TB, i\sqrt{\frac{\tau}{2}}\tilde{Z}_{n+1}\tilde{Z}_{n+1}^TB\right]. \tag{3.10}$$

Note that $Z_{n+1} \in \mathbb{C}^{d\times(k+q+2m)}$.

We can see that the number of columns of the matrix Z_{n+1} will increase as the time integration goes forward. Therefore, the elimination of redundant information in terms of a column compression technique is necessary. Note that the matrix Z_{n+1} is complex, so we need to use a column compression method for complex matrix. In this paper we apply the RRQR based column compression for complex data, see [24] for details.

RRQR based column compression for complex data:

- Compute $Z = QR$ with $Q \in \mathbb{C}^{d\times p}$, $Q^H Q = I_p$, $R \in \mathbb{C}^{p\times p}$.
- Compute a decomposition $RR^T = V\Lambda V^T$ with $V \in \mathbb{C}^{p\times p}$, $V^T V = I_p$ and a diagonal matrix $\Lambda \in \mathbb{C}^{p\times p}$ with diagonal entries $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_p|$.
- Set the compressed factor $Z_r := QV_r\Lambda_r^{\frac{1}{2}} \in \mathbb{C}^{d\times r}$, where $V_r = V(:, 1:r)$, $\Lambda_r = \Lambda(1:r, 1:r)$ and $r < p$ is the required rank.

If $r = p$, it follows from the above procedure that

$$Z_r Z_r^T = QV_r\Lambda_r^{\frac{1}{2}}\Lambda_r^{\frac{1}{2}}V_r^T Q^T = QV_r\Lambda_r V_r^T Q^T = QV\Lambda V^T Q^T = QRR^T Q^T = ZZ^T.$$

Then we have that $Z_r Z_r^T \approx ZZ^T$ for $r < p$. We remark that the eigendecomposition (e.g. in MATLAB) for the complex symmetric matrix RR^T in general does not lead to $RR^T = V\Lambda V^T$. We refer the readers to [24] for details on practical computation of the eigendecomposition $RR^T = V\Lambda V^T$ in the above procedure.

In Algorithm 3 we present the proposed low-rank MDS method for the solution of DREs (3.6).

Algorithm 3 Low-rank MDS method for DREs (3.6)

Require: Low-rank factors Z_0, C, B such that $X_0 = \hat{Z}_0\hat{Z}_0^T$, $Q = C^T C$, $P = BB^T$.

Ensure: $(\hat{Z}_{n+1}, t_{n+1})$ such that $X_{n+1} \approx \hat{X}_{n+1} := \hat{Z}_{n+1}\hat{Z}_{n+1}^T$.

1: Set $\tau = (T - t_0)/N$ and $t_n = t_0 + \tau n$;
2: LU factorize $I - \frac{1}{2}\tau A := LU$;
3: **for** $n = 0$ **to** $N - 1$ **do**
4: Set $\check{Z}_{n+1} = \left[\sqrt{\tau}C^T, \left(\frac{3}{2}I + \tau A\right)\hat{Z}_n, i\sqrt{\tau}\hat{Z}_n\hat{Z}_n^T B, i\left(\frac{1}{2}I + \tau A\right)\hat{Z}_n, i\hat{Z}_n\right]$;
5: Solve $LU\bar{Z}_{n+1} = \left[\sqrt{\tau}C^T, \left(I + \frac{1}{2}\tau A\right)\hat{Z}_n, i\sqrt{\frac{\tau}{2}}\hat{Z}_n\hat{Z}_n^T B, i\sqrt{\frac{\tau}{2}}\check{Z}_{n+1}\check{Z}_{n+1}^T B\right]$;
6: Column-compress $\hat{Z}_{n+1} \approx \bar{Z}_{n+1}$ by, e.g., RRQR;
7: **end for**

Theorem 3.2. *Assume that Assumption 1 holds and the column compression error in Algorithm 3 satisfies* $\|\bar{Z}_{n+1} - \hat{Z}_{n+1}\| \le \varepsilon$, $n = 0, 1, \ldots, N-1$. *Then the numerical solution* $\hat{X}_n$ *of the low-rank MDS algorithm for DREs (3.6) satisfies*

$$\|X(t_n) - \hat{X}_n\| \le \left(e^{(T-t_0)\alpha} - 1\right)\frac{c_1\tau^2}{\alpha} + \left(e^{(T-t_0)\beta/\sqrt{\tau}} - 1\right)\frac{c_2\varepsilon}{\beta\sqrt{\tau}}, \quad n = 0, 1, \ldots, N,$$

where α, β, c_1, c_2 *are positive constants.*

Proof. From Algorithm 3, we have

$$\check{Z}_{n+1} = \left[\sqrt{\tau}C^T, \left(\frac{3}{2}I + \tau A\right)\hat{Z}_n, i\sqrt{\tau}\hat{Z}_n\hat{Z}_n^T B, i\left(\frac{1}{2}I + \tau A\right)\hat{Z}_n, i\hat{Z}_n\right], \tag{3.11}$$

$$\bar{Z}_{n+1} = \left(I - \frac{1}{2}\tau A\right)^{-1}\left[\sqrt{\tau}C^T, \left(I + \frac{1}{2}\tau A\right)\hat{Z}_n, i\sqrt{\frac{\tau}{2}}\hat{Z}_n\hat{Z}_n^T B, i\sqrt{\frac{\tau}{2}}\check{Z}_{n+1}\check{Z}_{n+1}^T B\right]. \tag{3.12}$$

Subtracting (3.11) from (3.9) and taking norms we have

$$\begin{aligned}
\|\tilde{Z}_{n+1} - \check{Z}_{n+1}\| &= \| [0_{d\times q}, (\frac{3}{2}I + \tau A)(Z_n - \hat{Z}_n), i\sqrt{\tau}(Z_n Z_n^T - \hat{Z}_n\hat{Z}_n^T)B, \\
&\quad i(\frac{1}{2}I + \tau A)(Z_n - \hat{Z}_n), i(Z_n - \hat{Z}_n)] \| \\
&\le \underbrace{(\|\frac{3}{2}I + \tau A\|_2 + \|\frac{1}{2}I + \tau A\|_2 + 1)}_{=:\eta}\|Z_n - \hat{Z}_n\| \\
&\quad + \sqrt{\tau}\|B\|_2\|Z_n Z_n^T - \hat{Z}_n\hat{Z}_n^T\| \\
&\le (\eta + \sqrt{\tau}\|B\|_2(\|Z_n\| + \|\hat{Z}_n\|))\|Z_n - \hat{Z}_n\| \\
&\le \theta\|Z_n - \hat{Z}_n\|,
\end{aligned} \tag{3.13}$$

where we have used the inequality (see (3.4) for details)

$$\|Z_nZ_n^T - \hat{Z}_n\hat{Z}_n^T\| \le (\|Z_n\| + \|\hat{Z}_n\|)\|Z_n - \hat{Z}_n\| \tag{3.14}$$

and $\theta = \eta + \sqrt{\tau}\|B\|_2 \max_{n=0,1,\ldots,N}(\|Z_n\| + \|\hat{Z}_n\|)$.

Similarly, subtracting (3.12) from (3.10) and taking norms we have

$$\begin{aligned}
\|Z_{n+1} - \bar{Z}_{n+1}\| &= \| (I - \frac{1}{2}\tau A)^{-1}[0_{d\times q}, (I + \frac{1}{2}\tau A)(Z_n - \hat{Z}_n), \\
&\quad i\sqrt{\frac{\tau}{2}}(Z_nZ_n^T - \hat{Z}_n\hat{Z}_n^T)B, i\sqrt{\frac{\tau}{2}}(\tilde{Z}_{n+1}\tilde{Z}_{n+1}^T - \check{Z}_{n+1}\check{Z}_{n+1}^T)B] \| \\
&\le \|Z_n - \hat{Z}_n\| + \sqrt{\frac{\tau}{2}}\|B\|_2\|Z_nZ_n^T - \hat{Z}_n\hat{Z}_n^T\| \\
&\quad + \sqrt{\frac{\tau}{2}}\|B\|_2\|\tilde{Z}_{n+1}\tilde{Z}_{n+1}^T - \check{Z}_{n+1}\check{Z}_{n+1}^T\| \\
&\le (1 + \sqrt{\frac{\tau}{2}}\|B\|_2(\|Z_n\| + \|\hat{Z}_n\|))\|Z_n - \hat{Z}_n\| \\
&\quad + \sqrt{\frac{\tau}{2}}\|B\|_2(\|\tilde{Z}_{n+1}\| + \|\check{Z}_{n+1}\|)\|\tilde{Z}_{n+1} - \check{Z}_{n+1}\| \\
&\le (1 + \sqrt{\frac{\tau}{2}}\|B\|_2(\|Z_n\| + \|\hat{Z}_n\| + \theta(\|\tilde{Z}_{n+1}\| + \|\check{Z}_{n+1}\|)))\|Z_n - \hat{Z}_n\| \\
&\le (1 + \sqrt{\tau}\beta)\|Z_n - \hat{Z}_n\|,
\end{aligned} \tag{3.15}$$

where we have used the assumptions $\|(I - \frac{1}{2}\tau A)^{-1}(I + \frac{1}{2}\tau A)\|_2 \le 1$, $\|(I - \frac{1}{2}\tau A)^{-1}\|_2 \le 1$, (3.13) and (3.14), and the constant β is defined as $\beta := \frac{\sqrt{2}}{2}\|B\|_2(\max_{n=0,1,\ldots,N}(\|Z_n\| + \|\hat{Z}_n\|) + \theta \max_{n=0,1,\ldots,N}(\|\tilde{Z}_n\| + \|\check{Z}_n\|))$.

It then follows from (3.15) and the assumption $\|\bar{Z}_{n+1} - \hat{Z}_{n+1}\| \le \varepsilon$ that

$$\|Z_{n+1} - \hat{Z}_{n+1}\| \le \|Z_{n+1} - \bar{Z}_{n+1}\| + \|\bar{Z}_{n+1} - \hat{Z}_{n+1}\| \le (1 + \sqrt{\tau}\beta)\|Z_n - \hat{Z}_n\| + \varepsilon. \tag{3.16}$$

By using (3.14) and (3.16) we get

$$\begin{aligned}
\|X_{n+1} - \hat{X}_{n+1}\| &= \|Z_{n+1}Z_{n+1}^T - \hat{Z}_{n+1}\hat{Z}_{n+1}^T\| \\
&\le (\|Z_{n+1}\| + \|\hat{Z}_{n+1}\|)\|Z_{n+1} - \hat{Z}_{n+1}\| \\
&\le c_2((1 + \sqrt{\tau}\beta)\|Z_n - \hat{Z}_n\| + \varepsilon),
\end{aligned} \tag{3.17}$$

where $c_2 := \max_{n=0,1,\ldots,N}(\|Z_n\| + \|\hat{Z}_n\|)$.

Now from (2.16) we note that

$$\|X(t_{n+1}) - X_{n+1}\| \le (1 + \tau\alpha)\|X_n - X(t_n)\| + c_1\tau^3.$$

It then follows that

$$\begin{aligned}
\|X(t_{n+1}) - \hat{X}_{n+1}\| &= \|X(t_{n+1}) - X_{n+1} + X_{n+1} - \hat{X}_{n+1}\| \\
&\le \|X(t_{n+1}) - X_{n+1}\| + \|X_{n+1} - \hat{X}_{n+1}\| \\
&\le (1 + \tau\alpha)\|X_n - X(t_n)\| + c_1\tau^3 + c_2((1 + \sqrt{\tau}\beta)\|Z_n - \hat{Z}_n\| + \varepsilon).
\end{aligned}$$

By induction we obtain that

$$\begin{aligned}
\|X(t_n) - \hat{X}_n\| &\le (1 + \tau\alpha)^n\|X(t_0) - X_0\| + [(1 + \tau\alpha)^n - 1]\frac{c_1\tau^2}{\alpha} \\
&\quad + c_2\left[(1 + \sqrt{\tau}\beta)^n\|Z_0 - \hat{Z}_0\| + [(1 + \sqrt{\tau}\beta)^n - 1]\frac{\varepsilon}{\sqrt{\tau}\beta}\right] \\
&= [(1 + \tau\alpha)^n - 1]\frac{c_1\tau^2}{\alpha} + [(1 + \sqrt{\tau}\beta)^n - 1]\frac{c_2\varepsilon}{\sqrt{\tau}\beta} \\
&\le \left(e^{(T-t_0)\alpha} - 1\right)\frac{c_1\tau^2}{\alpha} + \left(e^{(T-t_0)\beta/\sqrt{\tau}} - 1\right)\frac{c_2\varepsilon}{\beta\sqrt{\tau}},
\end{aligned}$$

where we have used $\|X(t_0) - X_0\| = 0$, $\|Z_0 - \hat{Z}_0\| = 0$, $1 + \tau\alpha \le e^{\tau\alpha}$ and $n\tau \le (T - t_0)$, and the result follows. □

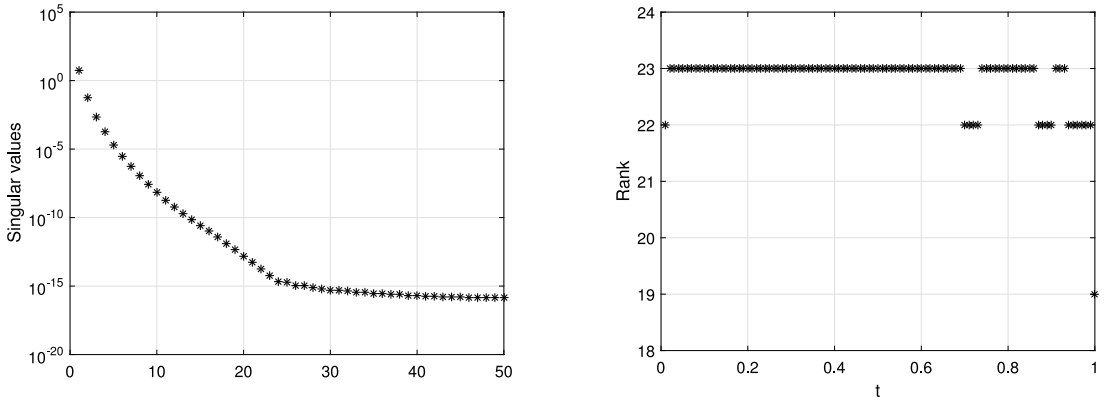

Fig. 4.1. Results for the DLE (4.2). Left: First 50 singular values of the reference solution at $T = 1$. Right: Rank of the reference solution vs time t.

4. Numerical experiments

In this section, we provide some numerical results for the MDS scheme (2.4) and its low-rank versions. Here, we mainly focus on the convergence properties of the (low-rank) MDS scheme for DLEs and DREs. All experiments are implemented using MATLAB.

Example 1. In this example we consider the heat equation

$$\frac{\partial u}{\partial t} = \triangle u$$

on $[0, 1]^2$ with homogeneous Dirichlet boundary conditions. Let us discretize the spatial domain by a rectangular mesh which is uniform in each direction as follows: $(x_i, y_j) = (ih, jh)$ for $0 \le i, j \le d + 1$ with $h = 1/(d + 1)$. Let $u_{i,j} = u_{i,j}(t) \approx u(t, x_i, y_j)$ for $0 \le i, j \le d + 1$ denote the numerical solution. By using the standard central difference discretization in both spatial directions, we obtain the following semi-discretized system

$$v' = (I_d \otimes A + A \otimes I_d)v, \tag{4.1}$$

where

$$v = [u_{1,1}, \ldots, u_{d,1}, u_{1,2}, \ldots, u_{d,2}, \ldots, u_{1,d}, \ldots, u_{d,d}]^T,$$

and A is the 1-dimensional discretization of the Laplacian and $A = \frac{1}{h^2}\text{tridiag}(1, -2, 1) \in \mathbb{R}^{d\times d}$. Now, we denote

$$X = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,d} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ u_{d,1} & u_{d,2} & \cdots & u_{d,d} \end{bmatrix} \in \mathbb{R}^{d\times d}.$$

By using the properties of the Kronecker product, the semi-discretization vector system (4.1) can be equivalently written as the DLE

$$\dot{X} = AX + XA^T.$$

In this example we consider the DLE

$$\dot{X} = AX + XA^T + \sin(\pi t)Q, \tag{4.2}$$

where $Q = C^T C$. The matrix $C \in \mathbb{R}^{10\times d}$ is defined by taking 10 vectors $\{f_1, f_2, \ldots, f_{10}\}$, where

$$f_k(x) = 2\sin(10k\pi hx), \quad k = 1, \ldots, 10,$$

are evaluated at the grid points $\{x_j\}_{j=1}^d$. Here we take $d = 400$ and integration interval $[t_0, T] = [0, 1]$. The initial value X_0 is chosen to be zero matrix.

In Fig. 4.1, we show the first 50 singular values of a reference solution computed by the MDS scheme (2.4) with stepsize $\tau = 10^{-5}$ at $T = 1$. This solution is taken as the reference solution for all tests on the DLE (4.2). We also show in Fig. 4.1 the rank of the reference solution vs time t. Here we use the MATLAB functions *svd* and *rank*. It is shown that the solution of the DLE has low-rank behavior.

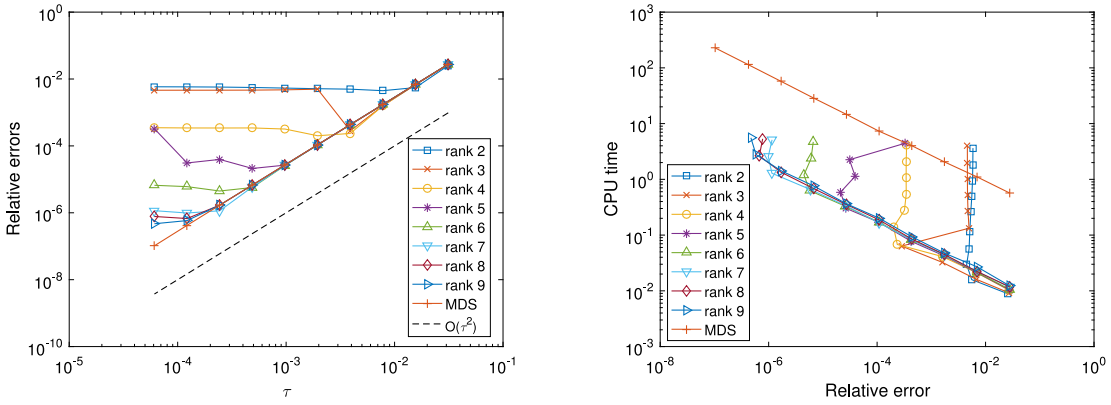

Fig. 4.2. Comparison between the MDS scheme described in Algorithm 1 (MDS) and the low-rank MDS method described in Algorithm 2 with different ranks (rank) for the DLE (4.2). Left: Relative errors in the Frobenius norm vs step size τ at $T = 1$. Right: Computing time vs step size at $T = 1$.

In Fig. 4.2 we compare the relative errors and CPU time obtained by the MDS scheme described in Algorithm 1 with the low-rank MDS method described in Algorithm 2. The relative error is defined by $\frac{\|X-X_{ref}\|}{\|X_{ref}\|}$ in Frobenius norm, where X is the numerical solution and X_{ref} is the reference solution. The CPU time (in seconds) is estimated using the timing functions *tic*/*toc*. We observe that the MDS scheme achieves second-order convergence rate. We also observe that the error of the low-rank MDS method includes two different contributions, the error due to the discretization and the error due to the low-rank approximation. When the error due to the low-rank approximation is not dominant, the expected second-order approximation accuracy is achieved. If the low-rank approximation error is dominant, decreasing the step size will not make the solution more accurate. These observations are in accordance with the convergence result stated in Theorem 3.1. It is also shown that a stagnation of the error around certain values depends on the rank. By comparing with Fig. 4.1 on the left, we note that these errors can be related to the magnitude of the first singular value discarded in the low-rank approximation procedure. Furthermore, Fig. 4.2 on the right shows that the proposed low-rank MDS method outperforms the full-rank MDS scheme in terms of computational time.

However, we also notice that the error term $\frac{\varepsilon}{\tau}$ that appears in Theorem 3.1 cannot be observed in Fig. 4.2 on the left. When the low-rank approximation error is dominant, we observe that the error does not go up much with decreasing step size which would be suggested by $\frac{\varepsilon}{\tau}$. It seems that the error bound in Theorem 3.1 is not optimal.

In Fig. 4.2 on the left, we also see that the error for the case of rank 3 is almost the same as for the case of rank 2 when the step size τ is small enough. This error behavior may due to the fact that the QR factorization used in RRQR column compression algorithm is not accurate enough for rank-deficient or ill-conditioned matrices. We remark that singular value decomposition (SVD) is more accurate but slower than QR factorization for ill-conditioned matrices. We refer the interested readers to [24] for the description of Rank-Revealing SVD based column compression algorithm. In general, the choice of algorithm depends on the relative importance of speed and reliability to the user.

In Fig. 4.3, we show the defects in symmetry and positive semidefiniteness of MDS and low-rank MDS methods for the DLE (4.2). The defects in symmetry and defects in positive semidefiniteness are defined by

$$d_{sym} = \frac{\|X - X^T\|}{\|X_{ref}\|}, \quad d_{psd} = \frac{\|X - \tilde{X}\|}{\|X_{ref}\|},$$

respectively, where $\tilde{X}$ denotes the nearest symmetric positive semidefinite matrix to X obtained by the algorithm proposed in [35]. We can see that the MDS scheme and its low-rank version preserve symmetry and positive semidefiniteness of the DLE. We remark that we only present the defects in positive semidefiniteness for the low-rank MDS method. Our numerical results show that the defects in symmetry for the low-rank MDS method are all zeros, so we choose to omit such results for brevity.

To see how big the splitting error is, we also present numerical results obtained by the midpoint scheme. The midpoint scheme for the DLE (4.2) is given by

$$\frac{X_{n+1} - X_n}{\tau} = \frac{A(X_n + X_{n+1})}{2} + \frac{(X_n + X_{n+1})A^T}{2} + \sin(\pi(t_n + \tau/2))Q.$$

At each time step, we need to solve an ALE of the form

$$(I - \tau A)X_{n+1} + X_{n+1}(I - \tau A)^T = 2X_n + \tau(AX_n + X_nA^T + 2\sin(\pi(t_n + \tau/2))Q).$$

Here we use MATLAB function *lyap* from the Control System Toolbox to compute the associated ALEs throughout this section.

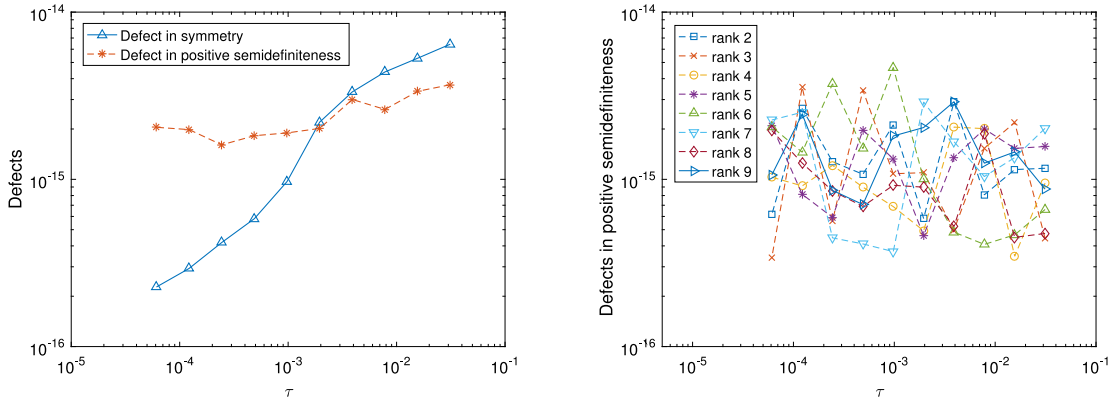

Fig. 4.3. Results for the DLE (4.2). Left: Defects in symmetry and in positive semidefiniteness for MDS scheme vs step size at $T = 1$. Right: Defects in positive semidefiniteness for low-rank MDS method vs step size at $T = 1$.

Table 1
Comparison of error results for the DLE (4.2) with the full-rank MDS scheme and the midpoint scheme.

τ	MDS		Midpoint	
	Error	Order	Error	Order
2^{-5}	2.7697e−2	–	1.9716e−3	–
2^{-6}	6.9377e−3	1.9972	4.9236e−4	2.0016
2^{-7}	1.7352e−3	1.9994	1.2305e−4	2.0004
2^{-8}	4.3386e−4	1.9998	3.0760e−5	2.0002
2^{-9}	1.0846e−4	2.0001	7.6879e−6	2.0004
2^{-10}	2.7114e−5	2.0001	1.9201e−6	2.0014
2^{-11}	6.7764e−6	2.0004	4.7818e−7	2.0056
2^{-12}	1.6920e−6	2.0018	1.1771e−7	2.0224
2^{-13}	4.2086e−7	2.0073	2.7559e−8	2.0946
2^{-14}	1.0308e−7	2.0296	5.1779e−9	2.4121

In Table 1, we report the relative errors at $T = 1$ and the experimental order of accuracy for the full-rank MDS scheme and the midpoint scheme. Note that the error of the full-rank MDS scheme is larger than the error of the midpoint scheme for fixed step size τ. The error constant of the full-rank MDS scheme is approximately 14 times larger than that of the midpoint scheme for this example.

Example 2 ([9])**.** Now we consider a problem that arises in a linear-quadratic control problem of a one-dimensional heat flow, see [9] for details. The associated DRE is as follows

$$\dot{X}(t) = A^T X(t) + X(t)A + C^T C - X(t)BB^T X(t). \tag{4.3}$$

The system matrices are given by

$$A = M^{-1}K, \quad B = M^{-1}b, \quad C = c^T,$$

where

$$K = -\frac{d+1}{100}\text{tridiag}(-1, 2, -1) \in \mathbb{R}^{d\times d}, \quad M = \frac{1}{6(d+1)}\text{tridiag}(1, 4, 1) \in \mathbb{R}^{d\times d},$$
$$b = c = \frac{1}{d+1}\left[0, \ldots, 0, \frac{1}{2}, 1, \ldots, 1, \frac{1}{2}, 0, \ldots, 0\right]^T \in \mathbb{R}^{d\times 1}.$$

Here the positions of $\frac{1}{2(d+1)}$ in b and c are $0.2d$ and $0.3d$, respectively. We take $d = 200$, initial value $X_0 = 0$ and time interval $[t_0, T] = [0, 5]$ in this example. A reference solution is computed by the MDS scheme (2.4) with stepsize $\tau = 10^{-6}$ at $T = 5$.

In Fig. 4.4, we report the error behavior and CPU time for the MDS scheme in Algorithm 1 and its low-rank version in Algorithm 3 applied to the DRE (4.3). We notice that the expected second-order approximation accuracy is obviously achieved for the MDS scheme. The low-rank MDS scheme achieves second-order accuracy when the error due the low-rank approximation is not dominant. We also observe that the low-rank MDS method performs better in terms of CPU

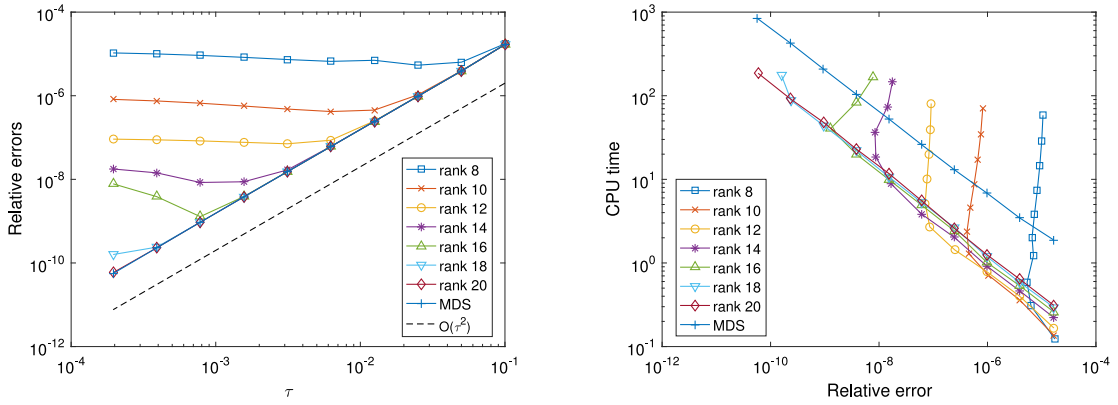

Fig. 4.4. Comparison between the MDS scheme described in Algorithm 1 (MDS) and the low-rank MDS method described in Algorithm 2 with different ranks (rank) for the DRE (4.3). Left: Relative errors in the Frobenius norm vs step size τ at $T = 5$. Right: Computing time vs step size at $T = 5$.

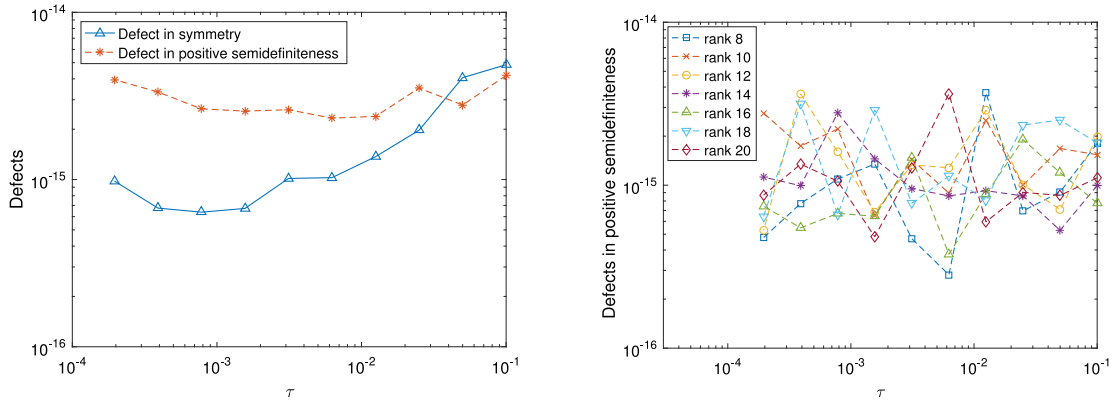

Fig. 4.5. Results for the DRE (4.3). Left: Defects in symmetry and in positive semidefiniteness for MDS scheme vs step size at $T = 5$. Right: Defects in positive semidefiniteness for low-rank MDS method vs step size at $T = 5$.

time provided that the required rank of the numerical solution is large enough, which numerically demonstrate the effectiveness of our proposed low-rank MDS algorithm. Similar to the DLE case, we observe that the error behavior of the low-rank MDS method is not totally in accordance with the convergence result stated in Theorem 3.2. The error term $\frac{\varepsilon e^{c/\sqrt{\tau}}}{\sqrt{\tau}}$ that appears in Theorem 3.2 cannot be observed in the numerical experiments. It seems that the error bound in Theorem 3.2 is not optimal. We remark that more in-depth error analysis of low-rank MDS schemes is beyond the scope of this paper and will be carried out in our future work.

In Fig. 4.5, we show the defects in symmetry and positive semidefiniteness of the MDS scheme and low-rank MDS scheme for the DRE (4.3). We again observe that both MDS scheme and its low-rank expression preserve symmetry and positive semidefiniteness of the DRE. Similar to Example 1, the computed defects in symmetry for the low-rank MDS method are all zeros and we omit such results for brevity.

To estimate the error constant of the full-rank MDS scheme, we also present results obtained by the second order Rosenbrock method [6]. Denote $F(X) = A^T X + XA + C^T C - XBB^T X$, then the second order Rosenbrock method for solving the DRE (4.3) is given by

$$\begin{cases} A_n^T K_1 + K_1 A_n = F(X_n), \\ A_n^T K_2 + K_2 A_n = F(X_n + \tau K_1) - 2K_1, \\ X_{n+1} = X_n + \frac{\tau}{2}(3K_1 + K_2), \end{cases} \tag{4.4}$$

where

$$A_n = \frac{1}{2} I - \gamma\tau(A - BB^T X_n)$$

and γ is a parameter. The second order Rosenbrock method is A-stable if $\gamma \geq \frac{1}{4}$ and L-stable when $\gamma = \frac{2+\sqrt{2}}{2}$. We remark that the focus of the numerical comparison is to show how big the splitting error is, so the general formulation of the

Table 2
Comparison of error results for the DRE (4.3) with the full-rank MDS scheme and second order Rosenbrock methods.

τ	MDS		Rosenbrock ($\gamma = \frac{1}{4}$)		Rosenbrock ($\gamma = \frac{2+\sqrt{2}}{2}$)	
	Error	Order	Error	Order	Error	Order
1/10	1.6758e−5	–	8.7255e−6	–	4.9431e−4	–
1/20	3.9045e−6	2.1016	2.1304e−6	2.0341	1.3300e−4	1.8940
1/40	9.7203e−7	2.0061	5.3329e−7	1.9981	3.4596e−5	1.9427
1/80	2.4311e−7	1.9994	1.3340e−7	1.9992	8.8298e−6	1.9702
1/160	6.0783e−8	1.9999	3.3357e−8	1.9997	2.2309e−6	1.9848
1/320	1.5190e−8	2.0005	8.3352e−9	2.0007	5.6073e−7	1.9922
1/640	3.7911e−9	2.0024	2.0786e−9	2.0036	1.4057e−7	1.9960
1/1280	9.4163e−10	2.0094	5.1433e−10	2.0148	3.5194e−8	1.9979
1/2560	2.2933e−10	2.0377	1.2342e−10	2.0591	8.8097e−9	1.9982
1/5120	5.6089e−11	2.0316	2.6569e−11	2.2158	2.2084e−9	1.9961

second order Rosenbrock scheme (4.4) is used here, more efficient reformulation of the scheme tailored to the case of autonomous DREs can be found in [6].

In Table 2, we report the relative errors at $T = 5$ and the experimental order of accuracy for the full-rank MDS scheme and second order Rosenbrock methods with different γ. Note that the error constant of the full-rank MDS scheme is approximately 1.8 times larger that of the second order Rosenbrock method with $\gamma = \frac{1}{4}$, while the error constant of the second order Rosenbrock method with $\gamma = \frac{2+\sqrt{2}}{2}$ is about 36 times larger than that of the full-rank MDS method.

We remark that the MATLAB function *lyap* from the Control System Toolbox is not efficient for solving large-scale ALEs. So we did not compare full-rank and low-rank MDS schemes with midpoint method and Rosenbrock method in terms of the computational cost. Comparison of efficiency for large-scale practical applications with low-rank MDS scheme and other low-rank methods will be carried out in our future research.

5. Conclusions

In this paper, we have applied the MDS scheme to solve a class of DMEs which includes DLEs and DREs. The implementation of the algorithm requires only the solution of a linear algebraic system with multiple right-hand sides in each time step. Theoretical and numerical results have shown that the proposed time-stepping method is second-order convergent and preserves symmetry and positive semidefiniteness of DLEs. Furthermore, we have proposed low-rank versions of the MDS scheme for solving DLEs and DREs, respectively. The convergence properties of the low-rank MDS methods have been studied. Numerical results indicate that the proposed low-rank MDS methods are effective.

Acknowledgments

The authors are very grateful to the referees for their constructive comments and valuable suggestions, which greatly improved the quality of this paper. This work was supported by the National Natural Science Foundation of China (Grant No. 11971085), the Natural Science Foundation Project of CQ CSTC, China (No. cstc 2018jcyjAX0113), the Scientific and Technological Research Program of Chongqing Municipal Education Commission, China (No. KJZD-M201800501), the Program of Chongqing Innovation Research Group Project in University, China (No. CXQT19018), and the Talent Project of Chongqing Normal University, China (No. 02030307-0054).

References

[1] H. Abou-Kandil, G. Freiling, V. Ionescu, G. Jank, Matrix Riccati Equations in Control and Systems Theory, Birkhäuser, Basel, 2003.
[2] A.C. Antoulas, Approximation of Large-Scale Dynamical Systems, SIAM, Philadelphia, 2005.
[3] A. Ichikawa, H. Katayama, Remarks on the time-varying H_∞ Riccati equations, Systems Control Lett. 37 (1999) 335–345.
[4] O.L.R. Jacobs, Introduction to Control Theory, Oxford Science Publications, Oxford, 1993.
[5] I.R. Petersen, V.A. Ugrinovskii, A.V. Savkin, Robust Control Design using H^∞ Methods, Springer, London, 2000.
[6] P. Benner, H. Mena, Rosenbrock methods for solving differential Riccati equation, IEEE Trans. Automat. Control 58 (2013) 2950–2957.
[7] P. Benner, H. Mena, Numerical solution of the infinite-dimensional LQR problem and the associated Riccati differential equations, J. Numer. Math. 26 (2018) 1–20.
[8] L. Dieci, Numerical integration of the differential Riccati equation and some related issues, SIAM J. Numer. Anal. 29 (1992) 781–815.
[9] H. Mena, Numerical Solution of Differential Riccati Equations Arising in Optimal Control Problems for Parabolic Partial Differential Equations (Ph.D. thesis), Escuela Politecnica Nacional, 2007.
[10] P. Benner, N. Lang, Peer methods for the solution of large-scale differential matrix equations, 2018, arXiv preprint arXiv:1807.08524.
[11] E. Hansen, T. Stillfjord, Convergence analysis for splitting of the abstract differential Riccati equation, SIAM J. Numer. Anal. 52 (2014) 3128–3139.
[12] A. Målqvist, A. Persson, T. Stillfjord, Multiscale differential Riccati equations for linear quadratic regulator problems, SIAM J. Sci. Comput. 40 (2018) A2406–A2426.
[13] H. Mena, A. Ostermann, L. Pfurtscheller, C. Piazzola, Numerical low-rank approximation of matrix differential equations, J. Comput. Appl. Math. 340 (2018) 602–614.

[14] A. Ostermann, C. Piazzola, H. Walach, Convergence of a low-rank Lie-Trotter splitting for stiff matrix differential equations, SIAM J. Numer. Anal. 57 (2019) 1947–1966.
[15] T. Stillfjord, Low-rank second-order splitting of large-scale differential Riccati equations, IEEE Trans. Automat. Control 60 (2015) 2791–2796.
[16] T. Stillfjord, Adaptive high-order splitting schemes for large-scale differential Riccati equations, Numer. Algorithms 78 (2018) 1129–1151.
[17] A. Koskela, H. Mena, A structure preserving Krylov subspace method for large scale differential Riccati equations, 2017, arXiv:1705.07507v1 / [math.NA].
[18] A.C. Antoulas, D.C. Sorensen, Y. Zhou, On the decay rate of Hankel singular values and related issues, Systems Control Lett. 46 (2002) 323–342.
[19] L. Grasedyck, Existence of a low rank or H-matrix approximant to the solution of a Sylvester equation, Numer. Linear Algebra Appl. 11 (2004) 371–389.
[20] T. Penzl, Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case, Systems Control Lett. 40 (2000) 139–144.
[21] T. Stillfjord, Singular value decay of operator-valued differential Lyapunov and Riccati equations, SIAM J. Control Optim. 56 (2018) 3598–3618.
[22] N. Truhar, K. Veselić, Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix, Systems Control Lett. 56 (2007) 493–503.
[23] N. Lang, H. Mena, J. Saak, An LDL^T factorization based ADI algorithm for solving large scale differential matrix equations, Proc. Appl. Math. Mech. 14 (2014) 827–828.
[24] N. Lang, H. Mena, J. Saak, On the benefits of the LDL^T factorization for large-scale differential matrix equation solvers, Linear Algebra Appl. 480 (2015) 44–71.
[25] N. Lang, Numerical Methods for Large-Scale Linear Time-Varying Control Systems and Related Differential Matrix Equations (Dissertation), Technische Universität Chemnitz, Germany, 2017.
[26] P. Benner, Z. Bujanović, P. Kürschner, J. Saak, RADI: a low-rank ADI-type algorithm for large scale algebraic Riccati equations, Numer. Math. 138 (2018) 301–330.
[27] V. Simoncini, Computational methods for linear matrix equations, SIAM Rev. 58 (2016) 377–441.
[28] A. Arrarás, K.J. in't Hout, W. Hundsdorfer, L. Portero, Modified Douglas splitting methods for reaction–diffusion equations, BIT 57 (2017) 261–285.
[29] J. Douglas, Alternating direction methods for three space variable, Numer. Math. 4 (1962) 41–63.
[30] J. Douglas, J.E. Gunn, A general formulation of alternating direction methods, Numer. Math. 6 (1964) 428–453.
[31] W. Hundsdorfer, J.G. Verwer, Numerical Solution of Time-Depenedent Advection-Diffusion-Reaction Equations, Springer, Berlin, 2003.
[32] W. Hundsdorfer, Accuracy and stability of splitting with stabilizing corrections, Appl. Numer. Math. 42 (2002) 213–233.
[33] R. Horn, C. Johnson, Topics in Matrix Analysis, Cambridge University Press, Cambridge, 1991.
[34] L. Dieci, T. Eirola, Positive definiteness in the numerical solution of Riccati differential equations, Numer. Math. 67 (1994) 303–313.
[35] N.J. Higham, Computing a nearest symmetric positive semidefinite matrix, Linear Algebra Appl. 103 (1988) 103–118.