Peter Benner* and Hermann Mena

Numerical solution of the infinite-dimensional LQR problem and the associated Riccati differential equations

https://doi.org/10.1515/jnma-2016-1039
Received April 26, 2016; revised April 28, 2016; accepted May 08, 2016

Abstract: The numerical analysis of linear quadratic regulator design problems for parabolic partial differential equations requires solving Riccati equations. In the finite time horizon case, the Riccati differential equation (RDE) arises. The coefficient matrices of the resulting RDE often have a given structure, e.g., sparse, or low-rank. The associated RDE usually is quite stiff, so that implicit schemes should be used in this situation. In this paper, we derive efficient numerical methods for solving RDEs capable of exploiting this structure, which are based on a matrix-valued implementation of the BDF and Rosenbrock methods. We show that these methods are suitable for large-scale problems by working only on approximate low-rank factors of the solutions. We also incorporate step size and order control in our numerical algorithms for solving RDEs. In addition, we show that within a Galerkin projection framework the solutions of the finite-dimensional RDEs converge in the strong operator topology to the solutions of the infinite-dimensional RDEs. Numerical experiments show the performance of the proposed methods.

Keywords: Riccati differential equation, Rosenbrock method, BDF method, LQR problem, parabolic control problem

Classification: 65L06, 65N12, 49N05, 93C20, 93D15

1 Introduction

The Riccati differential equation (RDE) is one of the most deeply studied nonlinear matrix differential equations arising in optimal control, optimal filtering, $\mathbf{H}_\infty$ control of linear-time varying systems, differential games, etc. [1, 42]. In the literature, there is a large variety of approaches to compute the solution of the RDE (see, e.g., [3, 22, 25]). However, none of these methods seems to be suitable for large-scale control problems, since the computational effort grows like n^3, where n is the dimension of the state of the control system. In this paper, we consider the numerical solution of large-scale RDEs arising in optimal control problems for linear parabolic partial differential equations. A variational formulation of the equation leads to an abstract Cauchy problem for a linear evolution equation of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \qquad \mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{H}, \quad \mathbf{y} = \mathbf{C}\mathbf{x} \tag{1.1}$$

for linear operators $\mathbf{A} : \operatorname{dom}(\mathbf{A}) \subset \mathcal{H} \to \mathcal{H}$, $\mathbf{B} : \mathcal{U} \to \mathcal{H}$, $\mathbf{C} : \mathcal{H} \to \mathcal{Y}$; where the state space $\mathcal{H}$, the observation space $\mathcal{Y}$, and the control space $\mathcal{U}$ are assumed to be separable Hilbert spaces. Additionally, $\mathcal{U}$ is assumed to be finite-dimensional, i.e., there is only a finite number of independent control inputs to (1.1). Here $\mathbf{C}$ maps the states of the system into its outputs, i.e., $\mathbf{y} = \mathbf{C}\mathbf{x}$. We also assume that the cost functional is given in a quadratic form, i.e.,

$$J(\mathbf{x}_0, \mathbf{u}) = \frac{1}{2}\int_0^{T_f} \langle \mathbf{x}, \mathbf{Q}\mathbf{x}\rangle_{\mathcal{H}} + \langle \mathbf{u}, \mathbf{R}\mathbf{u}\rangle_{\mathcal{U}}\, \mathrm{d}t + \langle \mathbf{x}_{T_f}, \mathbf{G}\mathbf{x}_{T_f}\rangle_{\mathcal{H}} \tag{1.2}$$

***Corresponding author: Peter Benner,** Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany. Email: benner@mpi-magdeburg.mpg.de
Hermann Mena, University of Innsbruck, Department of Mathematics, A-6020 Innsbruck, Austria.

where $\mathbf{Q}$, $\mathbf{G}$ are self-adjoint operators on the state space $\mathcal{H}$, $\mathbf{R}$ is a self-adjoint positive definitive operator on the control space $\mathcal{U}$, $\mathbf{x}_{T_f}$ denotes $\mathbf{x}(\cdot, T_f)$ and $T_f < \infty$. Usually, only a few measurements of the state are available as the outputs of the system, the operator $\mathbf{Q} := \mathbf{C}^*\tilde{\mathbf{Q}}\mathbf{C}$ generally is only positive semidefinite as well as $\mathbf{G}$. If the standard assumptions are:

- $\mathbf{A}$ is the infinitesimal generator of a strongly continuous semigroup $T(t)$;
- $\mathbf{B}$, $\mathbf{C}$ are linear bounded operators;
- for every initial value there exists an admissible control $\mathbf{u} \in L^2(0, \infty; \mathcal{U})$;

then, the solution of the abstract LQR problem can be obtained analogously to the finite-dimensional case [24, 27, 36, 50] as a feedback control

$$\mathbf{u}_*(t) = -\mathbf{R}^{-1}\mathbf{B}^*\mathbf{X}(t)\mathbf{x}_*(t) \tag{1.3}$$

where $\mathbf{X}(t)$ represents the unique nonnegative solution of the differential operator Riccati equation

$$\dot{\mathbf{X}}(t) = -(\mathbf{C}^*\mathbf{Q}\mathbf{C} + \mathbf{A}^*\mathbf{X}(t) + \mathbf{X}(t)\mathbf{A} - \mathbf{X}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{X}(t)) \tag{1.4}$$

with terminal condition $\mathbf{X}(T_f) = \mathbf{G}$. The rather restrictive assumption that $\mathbf{B}$ is bounded can be weakened [35, 36, 43].

In order to solve the infinite-dimensional LQR problem numerically, we use a Galerkin projection of the variational formulation onto a finite-dimensional space $\mathcal{H}^N$ spanned by a finite set of basis functions and then solve the discrete problem. Hence, we need to solve the large-scale RDEs resulting from the semi-discretization. The task of solving large-scale RDEs also arise in nonlinear optimal control problems in the context of receding horizon and model predictive control. There, linear problems have to be solved on certain time frames [30–32]. We provide a convergence theory for the approximation of the solution to the infinite-dimensional RDE by solutions of finite-dimensional RDEs following the seminal work by Gibson [27] and Banks and Kunisch [5]. This part of the paper is an abbreviated and revised version of the results first presented in [14, 38]. Similar results have recently also been reported in [49].

Typically, the coefficient matrices of the RDE have a given structure, e.g. sparse, symmetric, or low-rank. Moreover, we expect to treat stiff RDEs, so we will focus on methods that can efficiently deal with stiffness. We derive numerical methods capable of exploiting this structure. Particularly, we propose efficient matrix valued implementations of the backward differentiation formulae (BDF) and Rosenbrock type methods based on a low-rank approximation of the solution. Step size and order control strategies are also implemented based only on the solution factors.

This paper is organized as follows: in the next section, we present the approximation framework for the computation of Riccati operators. Then, efficient matrix valued algorithms of the BDF and Rosenbrock methods for large-scale DREs are proposed in Section 3. In Section 4 numerical examples show the performance of the proposed methods. Finally, conclusions and outlook are summarized in Section 5.

2 Theoretical aspects

The linear-quadratic control problem for infinite-dimensional systems has been deeply studied [18, 19, 36]. Particularly, approximation schemes for Riccati equations in infinite-dimensional spaces have been proposed in the last decades. Chronologically, the first reference is Gibson [27], who presented an approximation technique to reduce the inherently infinite-dimensional problems to finite-dimensional ones using Riccati integral equations. The result proposed by Gibson requires the approximating problems to be defined on the entire original state space, this leads to some technical difficulties. Assuming that the dynamics is modeled by an analytic semigroup, Banks/Kunisch [5] avoid these difficulties for the infinite-time horizon case. Moreover, convergence rates for some types of problems have been proved [33, 36].

In the following, we show that the solutions of the finite-dimensional RDEs converge in the strong operator topology to the solutions of the infinite-dimensional RDEs for the autonomous and the non-autonomous case. We use the same framework and assumptions as in [5, 27] and the generalization to the RDE case is to

some extent straightforward. Nevertheless, we emphasize that the available approximation results refer only to solutions of the Riccati integral equations. There are no explicit results for RDEs in the literature so far. Thus, we present the convergence results with short proofs for completeness and future reference.

2.1 Autonomous case

We consider the autonomous case, i.e., the case in which the coefficients of the partial differential equation are time-invariant.

Let $\mathcal{H}$ and $\mathcal{U}$ be Hilbert spaces and assume that $\mathbf{A}$: dom($\mathbf{A}$)$\subset \mathcal{H} \to \mathcal{H}$ is the infinitesimal generator of a strongly continuous semigroup $T(t)$ on $\mathcal{H}$, $B \in \mathcal{L}(\mathcal{U}, \mathcal{H})$.

We consider a control system in $\mathcal{H}$ given by (1.1) and the cost functional (1.2). We assume that (1.1) has a unique solution for a given $\mathbf{u}$ which is true, for example under the assumptions given in Section 1 (see [24, 36]). Here $\mathbf{Q} := \mathbf{C}^*\tilde{\mathbf{Q}}\mathbf{C}$, $\mathbf{G} \in \mathcal{L}(\mathcal{H})$, $\mathbf{R} \in \mathcal{L}(\mathcal{U})$ are self-adjoint with $\tilde{\mathbf{Q}} \geqslant 0$, $\mathbf{R} > 0$, $\mathbf{G} \geqslant 0$ and $\mathbf{x}_{T_f}$ denotes $\mathbf{x}(\cdot, T_f)$.

We will say that a function $\mathbf{u} \in L^2(0, T_f; \mathcal{U})$ is an admissible control for the initial state $\mathbf{x}_0 \in \mathcal{H}$ if $J(x_0, \mathbf{u})$ in (1.2) is finite. Let us consider the operator differential Riccati equation, then a solution of (1.4) in the interval $[0, T_f]$ is an operator $\Pi(t)$ such that $\Pi(T_f) = \mathbf{G}$ and for all $\varphi, \psi \in$ dom(A), $\langle \varphi, \Pi(\cdot)\psi\rangle$ is differentiable in $[0, T_f]$ and satisfies the equation,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\langle \varphi, \Pi(t)\psi\rangle \;=\; &-(\langle \varphi, \mathbf{Q}\psi\rangle + \langle \mathbf{A}\varphi, \Pi(t)\psi\rangle + \langle \Pi(t)\varphi, \mathbf{A}\psi\rangle \\ &-\langle \Pi(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\Pi(t)\varphi, \psi\rangle)\end{aligned} \tag{2.1}$$

as is defined in [18, Def. 2.1, p. 142]. Note that any solution of (1.4) is self-adjoint, and that $\Pi(\cdot)$ is nonnegative if $\mathbf{G}$ is. In order to solve numerically the operator Riccati differential equation for practical problems, we have to find suitable finite-dimension approximations to its solution. Therefore, let $\mathcal{H}^N$, $N = 1, 2 \ldots$, be a sequence of finite-dimensional linear subspaces of $\mathcal{H}$ and $P^N : \mathcal{H} \to \mathcal{H}^N$ be the canonical orthogonal projections. Assume that $T^N(t)$ is a sequence of continuous semigroups on $\mathcal{H}^N$ with infinitesimal generator $A^N \in \mathcal{L}(\mathcal{H}^N)$. Given operators $B^N \in \mathcal{L}(\mathcal{U}, \mathcal{H}^N)$, G^N, $Q^N \in \mathcal{L}(\mathcal{H}^N)$, $G^N \geqslant 0$, we consider the family of linear-quadratic regulator problems on $\mathcal{H}^N$:

Minimize:

$$J(x_0^N, \mathbf{u}) := \int_0^{T_f} \langle x^N, Q^N x^N\rangle_{\mathcal{H}^N} + \langle \mathbf{u}, \mathbf{R}\mathbf{u}\rangle_{\mathcal{U}}\,\mathrm{d}t + \langle x_{T_f}^N, G^N x_{T_f}^N\rangle_{\mathcal{H}^N}. \tag{$\mathcal{R}^N$}$$

with respect to

$$\begin{aligned}\dot{x}^N(t) &= A^N x^N(t) + B^N\mathbf{u}(t), \quad t > 0\\ x^N(0) &= x_0^N := P^N\mathbf{x}_0.\end{aligned}$$

($\mathcal{R}^N$) is a linear regulator problem in the finite-dimensional state space $\mathcal{H}^N$. If $Q^N \geqslant 0$, $\mathbf{R} > 0$, then the optimal control for ($\mathcal{R}^N$) is given in feedback form by

$$u_*^N(t) = -\mathbf{R}^{-1}B^{N*}\Pi^N(t)x_*^N(t)$$

where $\Pi^N(t) \in \mathcal{L}(\mathcal{H}^N)$ is the unique nonnegative self-adjoint solution of the Riccati differential equation:

$$\begin{aligned}\dot{\Pi}^N(t) &= -(Q^N + A^{N*}\Pi^N(t) + \Pi^N(t)A^N - \Pi^N(t)B^N\mathbf{R}^{-1}B^{N*}\Pi^N(t))\\ \Pi^N(T_f) &= G^N\end{aligned} \tag{2.2}$$

and $x_*^N(t)$ is the corresponding solution of the state equation with $\mathbf{u}(t) = u_*^N(t)$, see [1].

Let us now consider a related family of regulator problems, in which the operators are defined on the whole space,

Minimize:

$$J(x_0^N, \mathbf{u}) := \int_0^{T_f} \langle x^N, \bar{\mathbf{Q}}^N x^N \rangle_{\mathcal{H}} + \langle \mathbf{u}, \mathbf{R}\mathbf{u} \rangle_{\mathcal{U}} \, \mathrm{d}t + \langle x_{T_f}^N, \bar{\mathbf{G}}^N x_{T_f}^N \rangle_{\mathcal{H}} \tag{$\bar{\mathcal{R}}^N$}$$

with respect to

$$\begin{aligned} \dot{x}^N(t) &= \bar{\mathbf{A}}^N x^N(t) + B^N u(t), \quad t > 0 \\ x^N(0) &= x_0^N := P^N \mathbf{x_0} \end{aligned}$$

where $\bar{\mathbf{G}}^N := G^N P^N$, $\bar{\mathbf{Q}}^N := Q^N P^N$, $\bar{\mathbf{A}}^N := A^N P^N$ on $\mathcal{H}$. The problem ($\bar{\mathcal{R}}^N$) is considered as a problem in $\mathcal{H}$ even though we note that $x^N(t) \in \mathcal{H}^N$ for each t, so that $\bar{\mathbf{Q}}^N x^N(t) = Q^N x^N(t)$ and $\bar{\mathbf{G}}^N x^N(t_f) = G^N x^N(t_f)$.

The optimal control is given in terms of the solution of

$$\begin{aligned} \dot{\bar{\Pi}}^N(t) &= -(\bar{\mathbf{Q}}^N + \bar{\mathbf{A}}^{N*} \bar{\Pi}^N(t) + \bar{\Pi}^N(t) \bar{\mathbf{A}}^N - \bar{\Pi}^N(t) B^N \mathbf{R}^{-1} B^{N*} \bar{\Pi}^N(t)) \\ \bar{\Pi}^N(T_f) &= \bar{\mathbf{G}}^N. \end{aligned} \tag{2.3}$$

Note that

$$\bar{\Pi}^N(t) = \Pi^N(t) P^N. \tag{2.4}$$

In fact, if in (2.2) we replace Q^N, A^N, and G^N by $Q^N P^N$, $A^N P^N$, and $G^N P^N$, respectively, then it can be considered as an equation on $\mathcal{H}$. Moreover, (2.3) and (2.2) are the same equation and $\Pi^N(t) P^N$ is an extension of $\Pi^N(t) \in \mathcal{L}(\mathcal{H}^N)$ to the whole space $\mathcal{H}$, so (2.4) holds.

The convergence result of this section follows from [27]. The difference here, similar to [5], is that each of the finite-dimensional approximation problems are defined in a subspace of the state space, whereas in [27] the approximation problems have to be defined in the entire state space. Then, the result is formulated using ($\mathcal{R}^N$) rather than ($\bar{\mathcal{R}}^N$). This avoids some technical difficulties as explained in [5].

We will assume, similar to [5, (H2)], for $N \to \infty$:

(H)
- (i) For all $\varphi \in \mathcal{H}$ it holds that $T^N(t) P^N \varphi \to T(t)\varphi$ uniformly on any bounded subinterval of $[0, T_f]$;
- (ii) For all $\varphi \in \mathcal{H}$ it holds that $T^N(t)^* P^N \varphi \to T(t)^* \varphi$ uniformly on any bounded subinterval of $[0, T_f]$;
- (iii) For all $v \in \mathcal{U}$ it holds $B^N v \to \mathbf{B} v$ and for all $\varphi \in \mathcal{H}$ it holds that $B^{N*} P^N \varphi \to \mathbf{B}^* \varphi$;
- (iv) For all $\varphi \in \mathcal{H}$ it holds that $Q^N P^N \varphi \to \mathbf{Q}\varphi$;
- (v) For all $\varphi \in \mathcal{H}$ it holds that $G^N P^N \varphi \to \mathbf{G}\varphi$.

Assumption (ii) implies that $P^N \varphi \to \varphi$ for all $\varphi \in \mathcal{H}$, in this sense the subspaces $\mathcal{H}^N$ approximate $\mathcal{H}$.

Theorem 2.1. *Let* (H) *hold, then for* $N \to \infty$

$$\begin{aligned} u^N &\to u \quad \text{uniformly on } [0, T_f] \\ x^N &\to x \quad \text{uniformly on } [0, T_f] \end{aligned}$$

and for $\varphi \in \mathcal{H}$,

$$\Pi^N(t) P^N \varphi \to \Pi(t)\varphi \quad \text{uniformly in } t \in [0, T_f]. \tag{2.5}$$

Here u^N, u, x^N, x *denote optimal controls and trajectories of the problems* ($\mathcal{R}^N$) *and the infinite dimensional problem, respectively.*

Proof. Let $\Pi(t)$ be the unique element of $\mathcal{B}_\infty(0, T_f; \mathcal{H}, \mathcal{H})$, the set of all such functions essentially bounded on $[0, T_f]$, which satisfies the Riccati integral equation,

$$\Pi(t)\varphi = T^*(T_f, t) \mathbf{G} T(T_f, t)\varphi + \int_t^{T_f} T^*(\eta, t)[\mathbf{Q}(\eta) - \Pi(\eta)\mathbf{B}(\eta)\mathbf{R}^{-1}\mathbf{B}^*(\eta)\Pi(\eta)] T(\eta, t)\varphi \, \mathrm{d}\eta$$

for $t \in [0, T_f]$ and $\varphi \in \mathcal{H}$. By calculations in [27, pp. 544–546], $\Pi(t)$ is also the unique solution of the Riccati integral equation of Curtain and Pritchard [23]:

$$\Pi(t)\varphi = S^*(T_f, t)\mathbf{G}S(T_f, t)\varphi + \int_t^{T_f} S^*(\eta, t)[\mathbf{Q}(\eta) + \Pi(\eta)\mathbf{B}(\eta)\mathbf{R}^{-1}\mathbf{B}^*(\eta)\Pi(\eta)]S(\eta, t)\varphi \, \mathrm{d}\eta$$

for $t \in [0, T_f]$ and $\varphi \in \mathcal{H}$, where

$$S(t, s)\varphi = T(t, s)\varphi - \int_t^{T_f} T(t, \eta)\mathbf{B}(\eta)\mathbf{R}^{-1}\mathbf{B}^*(\eta)\Pi(\eta)S(\eta, t)\varphi d\eta$$

for $s, t \in [0, T_f]$, $0 \leqslant s \leqslant t \leqslant T_f$ and $\varphi \in \mathcal{H}$. Thus, $\Pi(t)$ uniquely satisfies the infinite-dimensional Riccati differential equation (2.1). Moreover, let $\bar{\Pi}^N(t)$ be the Riccati operator related to the problem $(\bar{\mathcal{R}}^N)$. By (2.4), the theorem is a direct consequence of the result proposed in [27, Theorem 5.1, p. 560]. □

We point out that it is possible to prove an analogue to Theorem 2.1 without the requirement $\mathcal{H}^N \subseteq \mathcal{H}$. If we assume that $(\mathcal{H}, \|\cdot\|)$, $(\mathcal{H}^N, \|\cdot\|_N)$ are Hilbert spaces (in general $\mathcal{H}^N \nsubseteq \mathcal{H}$), with $T(t)$, $T^N(t)$ strongly continuous semigroups on $\mathcal{H}$ and $\mathcal{H}^N$, respectively, and if we modify hypotheses (H), for $N \to \infty$, to:

(0) There exist bounded linear operators $P^N : \mathcal{H} \to \mathcal{H}^N$ satisfying $\left\|P^N\varphi\right\|_N \to \|\varphi\|$ for all $\varphi \in \mathcal{H}$;
(i) There exist constants M, ω such that $\left\|T^N(t)\right\|_N \leqslant Me^{\omega t}$ for all N and for each $\varphi \in \mathcal{H}$, $\left\|T^N(t)P^N\varphi - P^N T(t)\varphi\right\|_N \to 0$ as $N \to \infty$, uniformly on any bounded subinterval of $[0, T_f]$;
(ii) For all $\varphi \in \mathcal{H}$ it holds $\left\|T^{N*}(t)P^N\varphi - P^N T^*(t)\varphi\right\|_N \to 0$ as $N \to \infty$, uniformly on any bounded subinterval of $[0, T_f]$;
(iii) For all $v \in \mathcal{U}$, the operators $\mathbf{B} \in \mathcal{L}(\mathcal{U}, \mathcal{H})$, $B^N \in \mathcal{L}(\mathcal{U}, \mathcal{H}^N)$ satisfy $\left\|B^N v - P^N\mathbf{B}v\right\|_N \to 0$ and for all $\varphi \in \mathcal{H}$ it holds that $\left\|B^{N*}P^N\varphi - \mathbf{B}^*\varphi\right\|_U \to 0$; (H')
(iv) There exist operators $Q^N \in \mathcal{L}(\mathcal{H}^N)$ with $\left\|Q^N\right\|_N$ $N = 1, 2, \ldots$, bounded and for all $\varphi \in \mathcal{H}$ it holds that $\left\|Q^N P^N\varphi - P^N\mathbf{Q}\varphi\right\|_N \to 0$;
(v) There exist operators $G^N \in \mathcal{L}(\mathcal{H}^N)$ with $\left\|G^N\right\|_N$ $N = 1, 2, \ldots$, bounded and for all $\varphi \in \mathcal{H}$ it holds that $\left\|G^N P^N\varphi - P^N\mathbf{G}\varphi\right\|_N \to 0$;
(vi) For all N, the operators Q^N, G^N are nonnegative self-adjoint.

Given these assumptions, we can state a similar result as in Theorem 2.1, where the convergence in (2.5) is attained in norm, i.e.,

$$\left\|\Pi^N(t)P^N\varphi - P^N\Pi(t)\varphi\right\|_N \to 0 \quad \text{uniformly in } t \in [0, T_f]. \tag{2.6}$$

The proof of this result follows very close to the one of Theorem 2.1 once an analogue to [27, Theorem 5.1, p. 560], which permits $\mathcal{H}^N \nsubseteq \mathcal{H}$, has been proven. Note that [27, Theorem 5.1, p. 560] relies directly on [27, Lemma 5.1, p. 560]. This lemma can be modified as follows.

Lemma 2.1. *Let X be a Banach space, let $\{X^N\}_{N\geqslant 2}$ be a sequence of Banach spaces and let $P^N : X \to X^N$ be bounded linear operators satisfying Assumption (H')(0). Let Ω be a compact subset of $\mathbb{R}^n$ and let $A(\cdot) : \Omega \to \mathcal{L}(X)$, and for $N \geqslant 2$, let $A_N(\cdot) : \Omega \to \mathcal{L}(X^N, X)$. Suppose that $\|A_N(\xi)\|$ is uniformly bounded in N and ξ, and that, for each $x \in X$, $A_N(\xi)P^N x$ converges to $P^N A(\xi)x$ uniformly in ξ. Let $g(\cdot) : \Omega \to X$ be continuous and suppose there is a sequence of functions $g_N(\cdot)$ which converge uniformly to $g(\cdot)$. Then, the sequence $\{A_N(\cdot)P^N g_N(\cdot)\}$ converges uniformly to $P^N A(\cdot)g(\cdot)$.*

Proof. Let $\xi \in \Omega$ and $\|\cdot\|_N$ a norm on X^N. Note that

$$\begin{aligned}\left\|A_N(\xi)P^N g_N(\xi) - P^N A(\xi)g(\xi)\right\|_N &\leqslant \left\|A_N(\xi)P^N g_N(\xi) - A_N(\xi)P^N g(\xi)\right\|_N + \left\|A_N(\xi)P^N g(\xi) - P^N A(\xi)g(\xi)\right\|_N \\ &\leqslant \left\|A_N(\xi)\right\| \left\|P^N\right\| \left\|g_N(\xi) - g(\xi)\right\|_X + \left\|A_N(\xi)P^N g(\xi) - P^N A(\xi)g(\xi)\right\|_N\end{aligned}$$

then, by the hypotheses assumed the lemma holds. □

The repeated application of Lemma 2.1, and Lemma 5.1 [27, p. 560] allows also to prove an analogous result for the non-autonomous case which permits $\mathcal{H}^N \not\subseteq \mathcal{H}$. This version of the theorem could be very useful for developing certain types of approximation schemes, e.g., finite differences or spectral methods.

2.2 Non-autonomous case

In this section we extend the approximation results to the non-autonomous case, i.e., the case in which partial differential equations with time-varying coefficients are considered. Thus, the system dynamics is modeled by an evolution operator.

Let $\mathcal{H}$ and $\mathcal{U}$ be real Hilbert spaces and consider an evolution process defined by

$$x(t) = U(t,s)x(s) + \int_0^t U(t,v)\mathbf{B}(v)\mathbf{u}(v)\,\mathrm{d}v \tag{2.7}$$

where $0 \leqslant s \leqslant t \leqslant T_f < \infty$, $U(\cdot,\cdot)$ is a strong evolution operator on $\mathcal{H}$, $\mathbf{u} \in L^2(0, T_f; \mathcal{U})$, $x_0 \in \mathcal{H}$, and $\mathbf{B} \in \mathcal{B}_\infty(0, T_f; \mathcal{H}, \mathcal{H})$.

Note that (2.7) can be differentiated using

$$\frac{\partial}{\partial t}\langle y, U(t,s)x\rangle = \langle y, \mathbf{A}(s)U(t,s)x\rangle, \quad x \in \mathcal{D}_{\mathbf{A}}, \quad y \in \mathcal{H}, \quad t > s$$

where $\mathbf{A}(\cdot)$ is the generator of $U(\cdot,\cdot)$ and $\mathcal{D}_{\mathbf{A}}$ is the domain, in which $U(\cdot,\cdot)$ is a mild evolution family. We use the integral form of (2.7) in our presentation following [23, 27]. We consider the cost functional

$$\mathcal{J}(\mathbf{u}, x_0) = \int_0^{T_f} (\langle x(s), \mathbf{Q}(s)x(s)\rangle + \langle \mathbf{u}(s), \mathbf{R}\mathbf{u}(s)\rangle)\,\mathrm{d}s + \langle x(T_f), \mathbf{G}x(T_f)\rangle$$

where $x(t)$ is given by (2.7), $\mathbf{G} \in \mathcal{L}(\mathcal{H})$ is self-adjoint and nonnegative, $\mathbf{Q} \in \mathcal{B}_\infty(0, T_f; \mathcal{H}, \mathcal{H})$, $\mathbf{R} \in \mathcal{B}_\infty(0, T_f; \mathcal{U}, \mathcal{U})$ and for each t, $\mathbf{Q}(t)$, $\mathbf{R}(t)$ are nonnegative and self-adjoint and $\mathbf{R}(t)$ satisfies

$$\langle y, \mathbf{R}(t)y\rangle \geqslant \mu \|y\|^2 \quad \text{a.e. for some } \mu > 0.$$

Then, the quadratic cost problem is:

$$\begin{gathered}\text{Find the optimal control } u_0 \in L^2(T; \mathcal{U}) \\ \text{which minimizes } \mathcal{J}(u; t_0, x_0).\end{gathered} \tag{$\mathcal{NAR}$}$$

Again let $\mathcal{H}^N$, $N = 1, 2 \ldots$, be a sequence of finite-dimensional linear subspaces of $\mathcal{H}$ and $P^N : \mathcal{H} \to \mathcal{H}^N$ be the corresponding canonical orthogonal projections. Assume that $\{U^N(\cdot,\cdot)\}$ is a sequence of evolution operators on $\mathcal{H}^N$ with generator $A^N(\cdot) \in \mathcal{L}(\mathcal{H}^N)$ and that $\{B^N(\cdot)\}$, $\{Q^N(\cdot)\}$, $\{\mathbf{R}^N(\cdot)\}$, and $\{G^N\}$ are sequences of operators in $\mathcal{B}_\infty(t_0, T; \mathcal{U}, \mathcal{H}^N)$, $\mathcal{B}_\infty(t_0, T; \mathcal{H}^N, \mathcal{H}^N)$, $\mathcal{B}_\infty(t_0, T; \mathcal{U}, \mathcal{U})$, and $\mathcal{L}(\mathcal{H}^N)$, respectively, with $Q^N(\cdot)$, $\mathbf{R}^N(\cdot)$, and G^N semidefinite and self-adjoint. As in the last section we consider the sequences of optimal control

problems corresponding to these sequences of operators. Suppose that, for each $\varphi \in \mathcal{H}$ and $v \in \mathcal{U}$,

$$\begin{array}{lll}
\text{(i)} & U^N(t,s)P^N\varphi \to U(t,s)\varphi & \text{strongly}, \quad t_0 \leqslant s \leqslant t \leqslant T \\
\text{(ii)} & U^{N*}(t,s)P^N\varphi \to U^*(t,s)\varphi & \text{strongly}, \quad t_0 \leqslant s \leqslant t \leqslant T \\
\text{(iii)} & B^N(t)v \to \mathbf{B}(t)v & \text{strongly a.e.}, \\
\text{(iv)} & B^{N*}(t)P^N\varphi \to \mathbf{B}^*(t)\varphi & \text{strongly a.e.}, \\
\text{(v)} & Q^N(t)P^N\varphi \to \mathbf{Q}(t)\varphi & \text{strongly a.e.}, \\
\text{(vi)} & \mathbf{R}^N(t)v \to \mathbf{R}(t)v & \text{strongly a.e.}, \\
\text{(vii)} & G^NP^N\varphi \to \mathbf{G}\varphi & \text{strongly}, \\
& \text{as } N \to \infty. &
\end{array} \tag{G'}$$

In addition we require

$$\left\|U^N(t,s)\right\|, \quad \left\|B^N\right\|_{\mathcal{B}_\infty}, \quad \left\|Q^N\right\|_{\mathcal{B}_\infty}, \quad \left\|\mathbf{R}^N\right\|_{\mathcal{B}_\infty}, \quad \left\|G^N\right\| \tag{G"}$$

to be uniformly bounded in N, t, and s and require a constant m such that for each N, $Q^N(t) \geqslant m > 0$ for almost all t.

We call the previous assumptions (G') and (G") because they are a slight modification of the hypothesis formulated by Gibson in [27]. Specifically, in (G') the evolution operators corresponding to the approximating problems are defined in subspaces of the original state space of the original problem, whereas in [27] they are defined in the whole space.

As before the subspaces $\mathcal{H}^N$ approximate $\mathcal{H}$ in the sense that $P^N\varphi \to \varphi$ for all $\varphi \in \mathcal{H}$.

Theorem 2.2. *Let* (G′) *and* (G″) *hold. For our sequence of control problems, denote the initial states by* $x^N(0)$, *and let* $x^N(0) \to x(0)$*; denote the optimal controls by* $u^N(\cdot)$, *the optimal trajectories by* $x^N(\cdot)$, *and the solutions of the differential Riccati equations by* $\Pi^N(\cdot)$. *For the problem* $(\mathcal{NAR})$, *denote the corresponding quantities by* $x(0)$, $u(\cdot)$, $x(\cdot)$, *and* $\Pi(\cdot)$. *Then we have*

$$\begin{array}{l}
u^N(t) \to u(t) \ \text{ strongly a.e. and in } L^2(0, T_f; \mathcal{U}) \\
x^N(t) \to x(t) \ \text{ strongly pointwise and in } L^2(0, T_f; \mathcal{H})
\end{array} \tag{2.8}$$

and for $\varphi \in \mathcal{H}$,

$$\Pi^N(t)P^N\varphi \to \Pi(t)\varphi \ \text{ strongly pointwise and in } L^2(0, T_f; \mathcal{H}). \tag{2.9}$$

If $U(\cdot,\cdot)$ *is strongly continuous and* $\mathbf{B}(\cdot)$, $\mathbf{B}^*(\cdot)$, $\mathbf{Q}(\cdot)$, *and* $\mathbf{R}(\cdot)$ *are piecewise strongly continuous, uniform convergence in* (G′) *implies uniform convergence in* (2.8)–(2.9).

Proof. As for the autonomous case the sequence of control problems are defined in subspaces of the original state space similar to $(\mathcal{R}^N)$. Let us denote these problems as $(\mathcal{NAR}^N)$. If we consider a related family of control problems $(\overline{\mathcal{NAR}^N})$ which are defined in the whole space analogous to $(\bar{\mathcal{R}}^N)$, and assuming similar arguments on $\Pi(\mathbf{t})$ to the ones in the proof of Theorem 2.1, the proof of Theorem 2.2 follows directly from [27, Theorem 5.1, p. 560]. □

Like in the autonomous case, it is possible to prove an analogue to Theorem 2.2 without the requirement $\mathcal{H}^N \subseteq \mathcal{H}$. Let us assume that $(\mathcal{H}, \|\cdot\|)$, $(\mathcal{H}^N, \|s\cdot\|_N)$ are Hilbert spaces (in general $\mathcal{H}^N \nsubseteq \mathcal{H}$), with $U(t,s)$, $U^N(t,s)$

strongly continuous evolution operators on $\mathcal{H}$ and $\mathcal{H}^N$, respectively. For this, we modify (G') to:

(0) There exist bounded linear operators $P^N : \mathcal{H} \to \mathcal{H}^N$ satisfying $\left\|P^N\varphi\right\|_N \to \|\varphi\|$ for all $\varphi \in \mathcal{H}$;
(i) There exist $M,\ \omega$ such that $\left\|U^N(t,s)\right\|_N \leqslant Me^{\omega(t-s)},\ t \geqslant s$ for all N and for each $\varphi \in \mathcal{H}$, $\left\|U(t,s)^N P^N\varphi - P^N U(t,s)\varphi\right\|_N \to 0$ as $N \to \infty$, uniformly on any bounded subinterval of $[0, T_f]$;
(ii) For all $\varphi \in \mathcal{H}$ it holds $\left\|U^{N*}(t,s)P^N\varphi - P^N U^*(t,s)\varphi\right\|_N \to 0$ as $N \to \infty$, uniformly on any bounded subinterval of $[0, T_f]$;
(iii) For all $v \in \mathcal{U}$, the operators $\mathbf{B} \in \mathcal{L}(\mathcal{U}, \mathcal{H})$, $B^N \in \mathcal{L}(\mathcal{U}, \mathcal{H}^N)$ satisfy $\left\|B^N v - P^N\mathbf{B}v\right\|_N \to 0$ and for all $\varphi \in \mathcal{H}$ it holds that $\left\|B^{N*}P^N\varphi - \mathbf{B}^*\varphi\right\|_U \to 0$;
(iv) There exist operators $Q^N \in \mathcal{L}(\mathcal{H}^N)$ with $\left\|Q^N\right\|_N$ $N = 1, 2, \ldots,$ bounded and for all $\varphi \in \mathcal{H}$ it holds that $\left\|Q^N P^N\varphi - P^N\mathbf{Q}\varphi\right\|_N \to 0$;
(v) There exist operators $G^N \in \mathcal{L}(\mathcal{H}^N)$ with $\left\|G^N\right\|_N$ $N = 1, 2, \ldots,$ bounded and for all $\varphi \in \mathcal{H}$ it holds that $\left\|G^N P^N\varphi - P^N\mathbf{G}\varphi\right\|_N \to 0$;
(vi) For all N, the operators $Q^N,\ G^N$ are nonnegative self-adjoint.

We can state a similar result as in Theorem 2.2, where the convergence is attained in norm. As in the previous section this can be proved as a consequence of the repeated application of Lemma 2.1 and Lemma 5.1 [27, p. 560].

Remark 2.1. The results in this subsection are particularly useful solving nonlinear problems in model predictive control and receding horizon context. There the LQG approach is applied to a linearization around a reference trajectory. This requires the solution of RDEs, in which the coefficient matrices are time dependent [7–9, 30, 32].

Remark 2.2. Note that the solution of the RDE is suboptimal in terms of the optimal cost which is of interest in applications. The optimal cost for the infinite $\bar{\mathbf{J}}$ and finite dimensional $\bar{J}_N$ control problems, can be found as

$$\bar{\mathbf{J}}(\mathbf{x}, \mathbf{u}) = \mathbf{x}_0{}^* \Pi(0)\mathbf{x}_0, \qquad \bar{J}_N(x, \mathbf{u}) = x_0^{N*}\Pi^N(0)x_0^N. \tag{2.10}$$

With an established convergence scheme, we will now discuss the numerical solution of finite-dimensional RDEs arising from the semi-discretizations obtained from the Galerkin projection framework. In the calculations shown later in Section 5, the Galerkin projections are obtained by using linear finite elements.

3 Numerical methods for large-scale RDEs

Although the RDEs have to be solved backward in time, making a change of variables we can solve RDEs forward in time and afterwards recover the original solution. Thus, let us consider time-varying symmetric RDEs of the form

$$\begin{aligned} \dot{X}(t) &= Q(t) + X(t)A(t) + A^T(t)X(t) - X(t)S(t)X(t) \\ X(t_0) &= X_0 \end{aligned} \tag{3.1}$$

where $t \in [t_0, t_f]$ and $Q(t)$, $A(t)$, $S(t)$, $\in \mathbb{R}^{n\times n}$, $X(t) \in \mathbb{R}^{m\times n}$. We assume that the coefficient matrices are piecewise continuous locally bounded matrix-valued functions, which ensure existence of the solution and uniqueness of (3.1), see, e.g., [1, Theorem 4.1.6].

Note also that using FEM, we obtain an linear system of the form

$$M\dot{x} = -Sx + Bu$$

where M is the positive definite mass matrix and S is the stiffness matrix corresponding to the FEM applied to the spatial differential operator. In many situations, S is symmetric positive definite and hence, $A := -M^{-1}S$ is stable, also $A := -M^{-1}B$. But of course, in computations we avoid forming A explicitly since A is usually a dense matrix. There are several approaches that can be used to treat this problem [6].

3.1 BDF methods

In the following we briefly describe the BDF methods for RDEs in matrix-valued form similar to [22, 25] and discuss an efficient implementation for large-scale problems following [13]. Let us define

$$F(t, X(t)) \equiv Q(t) + X(t)A(t) + A^T(t)X(t) - X(t)S(t)X(t). \tag{3.2}$$

The fixed-coefficients BDF methods applied to the RDE (3.1) yield

$$X_{k+1} = \sum_{j=1}^{p} -\alpha_{j+1}X_{k-j} + h\beta F(t_{k+1}, X_{k+1})$$

where h is the step size, $t_{k+1} = h + t_k$, $X_{k+1} \approx X(t_{k+1})$ and α_j, β are the coefficients for the p-step BDF formula (see, e.g., [4]). Hence, noting $Q_{k+1} \approx Q(t_{k+1})$, $A_{k+1} \approx A(t_{k+1})$, $S_{k+1} \approx S(t_{k+1})$, we obtain the Riccati-BDF difference equation

$$-X_{k+1} + h\beta(Q_{k+1} + A_{k+1}^T X_{k+1} + X_{k+1}A_{k+1} - X_{k+1}S_{k+1}X_{k+1}) - \sum_{j=1}^{p} \alpha_{j+1}X_{k-j} = 0.$$

Re-arranging terms, we see that this is an algebraic Riccati equation (ARE) for X_{k+1},

$$\begin{aligned} \left(h\beta Q_{k+1} - \textstyle\sum_{j=0}^{p-1} \alpha_j X_{k-j}\right) + \left(h\beta A_{k+1} - \tfrac{1}{2}I\right)^T X_{k+1} \\ + X_{k+1}\left(h\beta A_{k+1} - \tfrac{1}{2}I\right) - X_{k+1}(h\beta S_{k+1})X_{k+1} = 0 \end{aligned} \tag{3.3}$$

that can be solved via any method for AREs. In large-scale applications it is not possible to construct explicitly the matrices X_k, because they are in general dense. However, X_k is usually of low numerical rank [2, 28, 40, 45], i.e., it can be well approximated by a low-rank factor Z_k with $z_k \ll n$ for all times. Moreover, usually Q_k and S_k can also be represented in factored form. Thus, assuming that

$$\begin{aligned} Q_k &= C_k^T C_k, & C_k &\in \mathbb{R}^{p\times n} \\ S_k &= B_k B_k^T, & B_k &\in \mathbb{R}^{n\times m} \\ X_k &= Z_k Z_k^T, & Z_k &\in \mathbb{R}^{n\times z_k} \end{aligned} \tag{3.4}$$

the ARE (3.3) can be written as

$$\hat{C}_{k+1}^T\hat{C}_{k+1} + \hat{A}_{k+1}^T Z_{k+1}Z_{k+1}^T + Z_{k+1}Z_{k+1}^T\hat{A}_{k+1} - Z_{k+1}Z_{k+1}^T\hat{B}_{k+1}\hat{B}_{k+1}^T Z_{k+1}Z_{k+1}^T = 0 \tag{3.5}$$

where

$$\begin{aligned} \hat{A}_{k+1} &= h\beta A_{k+1} - \frac{1}{2}I \\ \hat{B}_{k+1} &= \sqrt{h\beta}B_{k+1} \\ \hat{C}_{k+1}^T &= \left[\sqrt{h\beta}C_{k+1}^T, \sqrt{-\alpha_1}Z_k, \ldots, \sqrt{-\alpha_p}Z_{k+1-p}\right]. \end{aligned}$$

If $z_k \ll n$ for all times and (3.5) can be solved efficiently by exploiting sparsity in A_{k+1} as well as the low-rank nature of the constant and quadratic terms, this can serve as the basis for a RDE solver for large-scale problems.

In our numerical implementation, we benefit from recent algorithmic progress in solving large-scale AREs and Lyapunov equations [10, 11]. There the main idea is to solve AREs using Newton's method as a one step iteration. It results in solving one Lyapunov equation in each step. The structure of the coefficient matrix in this equation has the form 'sparse + low-rank perturbation'. So, we need a solver, which exploits efficiently this structure. For a detailed explanation about the implementation of the BDF methods for large-scale RDEs we refer the reader to [13] and references therein.

3.1.1 Lyapunov equation solver

Efficient numerical methods for solving large and sparse Lyapunov equations have been proposed in recent years. A common approach is based on the low-rank alternating direction implicit (ADI) iteration [11, 37, 39]. In our implementation we use recent advances of ADI based solvers [10, 11]. The low-rank representation of the solution relies on the decay of the singular values. This phenomenon has been deeply studied and it is frequent in applications [2, 28, 40, 45]. Methods based on extended and rational Krylov subspace have also proved to be practical alternatives [47], however, we do not consider this approach here. A state-of-the-art survey of the methods is presented in [17].

3.1.2 Adaptive step size and order control

For varying the step size and order, an estimate of the error is needed. The local truncation error for the BDF methods can be written as:

$$h_k \dot{\omega}_k(t_k)[x_k, x_{k-1}, \ldots, x_{k-p}] \tag{3.6}$$

where $\omega_k(t) = \prod_{i=0}^{p}(t - t_{k-i})$, and $[x_k, x_{k-1}, \ldots, x_{k-p}]$ represents the divided differences, then

$$\dot{\omega}_k(t_k) = \prod_{i=1}^{p}(t_k - t_{k-i}) = \prod_{i=1}^{p}(h + \psi_{i-1}(k))$$

for $\psi_j(k) := t_k - t_{k-j}$, see [26]. This allow us to compute the error directly for low-rank factors of the solution of the RDEs, see Algorithm 3.3.

Note that, in addition the solution values at past times on an equidistant mesh are needed. For the BDF methods we can approximate these values using an interpolating polynomial described by Neville's algorithm, which in matrix valued form can be expressed as in Algorithm 3.1.

Algorithm 3.1 Neville's algorithm

Require: $\{(t_i, X_i)\}_{0 \leqslant i \leqslant n}$, $t_i \in I \subset \mathbb{R}$, $X_i \approx X(t_i) \in \mathbb{R}^{n \times n}$.

1: $T_{i,o} := X_i, \quad 0 \leqslant i \leqslant n.$

2: $T_{i,k} := \dfrac{(t - t_{i-k})T_{i,k-1} - (t - t_i)T_{i-1,k-1}}{t_i - t_{i-k}}, \quad 0 \leqslant i < k \leqslant n.$

Assuming that

$$X_i = Z_i Z_i^T, \qquad Z_i \in \mathbb{R}^{n \times z_i}$$

we get

$$\begin{aligned} Z_{i,k}Z_{i,k}^T &:= \frac{(t-t_{i-k})Z_{i,k-1}Z_{i,k-1}^T - (t-t_i)Z_{i-1,k-1}Z_{i-1,k-1}^T}{t_i - t_{i-k}} \\ &= \begin{bmatrix} \sqrt{\frac{t-t_{i-k}}{t_i-t_{i-k}}}Z_{i,k-1} & \sqrt{\frac{t-t_i}{t_{i-k}-t_i}}Z_{i-1,k-1} \end{bmatrix} \\ &\quad \times \begin{bmatrix} \sqrt{\frac{t-t_{i-k}}{t_i-t_{i-k}}}Z_{i,k-1} & \sqrt{\frac{t-t_i}{t_{i-k}-t_i}}Z_{i-1,k-1} \end{bmatrix}^T \end{aligned}$$

so that

$$Z_{i,k} = \begin{bmatrix} \sqrt{\frac{t-t_{i-k}}{t_i-t_{i-k}}}Z_{i,k-1} & \sqrt{\frac{t-t_i}{t_{i-k}-t_i}}Z_{i-1,k-1} \end{bmatrix}.$$

Hence Algorithm 3.1 can be written in terms of the low-rank factors LRFs (see Algorithm 3.2).

Algorithm 3.2 LRF Neville's algorithm

Require: $\{(t_i, Z_i)\}_{0\leqslant i\leqslant n}$, $t_i \in I \subset \mathbb{R}$ and $Z_i \approx Z(t_i) \in \mathbb{R}^{n\times z_i}$.

1: $Z_{i,o} := Z_i, \quad 0 \leqslant i \leqslant n.$

2: $Z_{i,k} := \begin{bmatrix} \sqrt{\frac{t-t_{i-k}}{t_i-t_{i-k}}}Z_{i,k-1} & \sqrt{\frac{t-t_i}{t_{i-k}-t_i}}Z_{i-1,k-1} \end{bmatrix}, \quad 0 \leqslant i < k \leqslant n.$

Since the size of $Z_{i,k}$ increases in every step, the computation becomes expensive. We can avoid the recursion formula expressing the final value given by the algorithm like

$$Z_{k,k} = \begin{bmatrix} \sqrt{\lambda_0}Z_{0,0} & \sqrt{\lambda_1}Z_{1,0} & \dots & \sqrt{\lambda_k}Z_{k,0} \end{bmatrix}.$$

For instance, if we consider $\{(t_i, Z_i)\}_{1\leqslant i\leqslant 2}$, then

$$Z_{2,2} = \begin{bmatrix} \sqrt{\alpha_{220}\alpha_{110}}Z_{0,0} & \sqrt{-(\alpha_{020}\alpha_{221}+\alpha_{220}\alpha_{010})}Z_{1,0} & \sqrt{\alpha_{020}\alpha_{121}}Z_{2,0} \end{bmatrix}$$

where

$$\alpha_{ijk} = \frac{t-t_i}{t_j-t_k}, \quad i,j,k = 0,1,2.$$

Algorithm 3.2 will in general generate complex factors. However, we can still get real factors as solutions of the RDE in every step rewriting

$$Z_{k,k} = [Z_p \quad \imath Z_n]$$

where Z_p and Z_n are formed by grouping the positive and negative λ's, respectively, and computing the operations involving $Z_{k,k}$ separately for Z_p and Z_n, i.e., never forming $Z_{k,k}$ explicitly.

Once the solution values at past times are approximated, we are ready to apply step size and order control. For this we need to compute local error estimators, this can be done using (3.6) and computing the divided differences directly for the factors, see Algorithm 3.3.

Algorithm 3.3 LRF divided differences

Require: $\{(t_i, Z_i)\}_{0\leqslant i\leqslant n}$, $t_i \in I \subset \mathbb{R}$ and $Z_i \approx Z(t_i) \in \mathbb{R}^{n\times z_i}$.

1: $Z_{i,o} := Z_i, \quad 0 \leqslant i \leqslant 0.$

2: $Z_{i,k} := \begin{bmatrix} \sqrt{\frac{1}{t_i-t_{i-k}}}Z_{i,k-1} & \sqrt{\frac{1}{t_{i-k}-t_i}}Z_{i-1,k-1} \end{bmatrix}, \quad 0 \leqslant i < k \leqslant 0.$

Analogous to Algorithm 3.2, Algorithm 3.3 can be implemented avoiding the recursive formula. Moreover, it generates, in general, complex factors which is not a problem here, because we are interested in the norm of the resulting factor to estimate the local truncation error using (3.6).

Another option to efficiently implement an adaptive algorithm is to consider a variable-coefficient formula of the method. Thus, changing the step size and order of the method is performed as for the one step methods for solving ODEs.

3.1.3 Variable-coefficient BDF methods

By using the variable-coefficient BDF methods (3.7), we avoid to compute the solution values at past times on an equidistant mesh. The application of this method to (3.1) yields an equation similar to (3.5) in which $\hat{A}_{k+1}$, $\hat{B}_{k+1}$ and $\hat{C}_{k+1}$ depend on $\tilde{\alpha}_i(h_n, h_{n-1}, \ldots, h_{n-k+1})$, $\tilde{\beta}(h_n, h_{n-1}, \ldots, h_{n-k+1})$. The computation of these coefficients is cheap and does not outweigh the iteration.

Working on unequally spaced meshes, we can derive the variable-coefficient BDF by rewriting the method as a general multistep like

$$\sum_{i=0}^{p} \tilde{\alpha}_i x_{k-i} = h_k \tilde{\beta} f(t_k, x_k) \tag{3.7}$$

where the coefficients $\tilde{\alpha}_i$, $\tilde{\beta}$ depend on the $p-1$ past steps, i.e.

$$\tilde{\alpha}_i = \tilde{\alpha}_i(h_k, h_{k-1}, \ldots, h_{k-p+1}), \quad \tilde{\beta} = \tilde{\beta}(h_k, h_{k-1}, \ldots, h_{k-p+1}).$$

The variable coefficients for the second and third order BDF methods can be found in [38, p. 50].

3.2 Rosenbrock methods

Linear multi-step methods require fewer function evaluations per step than one step methods, and they allow a more simple streamlined method design from the point of view of order and error estimation. However, the associated overhead is higher, e.g., for changing the step size. Runge–Kutta methods work well for the numerical solution of ODEs that are non-stiff. When stiffness becomes an issue: diagonally implicit Runge–Kutta methods or collocation methods offer an alternative to the BDF methods. In particular, linearly implicit one-step methods (better known as Rosenbrock methods) give satisfactory results [21, 29]. The idea of these methods can be interpreted as the application of one Newton iteration to each stage of an implicit Runge–Kutta method and the derivation of stable formulae by working with the Jacobian matrix directly within the integration formulae. In the following we focus on solving autonomous RDEs by an efficient implementation of Rosenbrock methods based on a low-rank version of the ADI iteration. We describe the linearly implicit Euler method and the second order method, the main ideas can be straightforwardly applied to higher order Rosenbrock methods.

3.2.1 Linearly implicit Euler method

The one-stage Rosenbrock method applied, as a matrix valued algorithm, to autonomous RDEs of the form (3.1) can be written as

$$\begin{aligned} \bar{A}_k^T K_1 + K_1 \bar{A}_k &= -F(X_k) - hF_{t_k} \\ X_{k+1} &= X_k + K_1 \end{aligned} \tag{3.8}$$

where $\bar{A}_k = A_k - R_k X_k - \frac{1}{2h} I$ and F are defined as in (3.2). Moreover, (3.8) can be re-written such that the next iterate is computed directly from the Lyapunov equation, i.e.,

$$\bar{A}_k^T X_{k+1} + X_{k+1} \bar{A}_k = -Q - X_k S X_k - \frac{1}{h} X_k. \tag{3.9}$$

The right-hand side of (3.9) is simpler to evaluate than the one in (3.8), so the implementation of (3.9) is more efficient [15]. If we assume,

$$\begin{aligned} Q &= C^T C, & C &\in \mathbb{R}^{p\times n} \\ S &= BB^T, & B &\in \mathbb{R}^{n\times m} \\ X_k &= Z_k Z_k^T, & Z_k &\in \mathbb{R}^{n\times z_k} \end{aligned} \tag{3.10}$$

with $p, m, z_k \ll n$ and denoting $N_k = [\, C^T \;\; Z_k(Z_k^T B) \;\; \sqrt{h^{-1}} Z_k \,]$, then the Lyapunov equation (3.9) results in

$$\bar{A}_k^T X_{k+1} + X_{k+1}\bar{A}_k = -N_k N_k^T \tag{3.11}$$

where $\bar{A}_k = A - B(Z_k(Z_k^T B))^T - \frac{1}{2h} I$. Observing that $\operatorname{rank}(N_k) \leqslant p + m + z_k \ll n$, we can use the modified version of the ADI iteration to solve (3.11). This will ensure low-rank factors Z_{k+1}, of X_{k+1}, such that $X_{k+1} \approx Z_{k+1} Z_{k+1}^T$, where $Z_{k+1} \in \mathbb{R}^{n\times z_{k+1}}$ with $z_{k+1} \ll n$. For an equidistant mesh the latter is sketched in Algorithm 3.4.

Algorithm 3.4 LRF linearly implicit Euler method

Require: $A \in \mathbb{R}^{n\times n}$, B, C, Z_0 satisfying (3.10), $t \in [a, b]$, and h step size.
Ensure: (Z_k, t_k) such that $X_k \approx Z_k Z_k^T$, $Z_k \in \mathbb{R}^{n\times z_i}$ with $z_i \ll n$.

1: $t_0 = a$.
2: **for** $k = 0$ to $\lceil \frac{b-a}{h} \rceil$ **do**
3: $\bar{A}_k = A - B(Z_k(Z_k^T B))^T - \frac{1}{2h} I$.
4: $N_k = [\, C^T \;\; Z_k(Z_k^T B) \;\; \sqrt{h^{-1}} Z_k \,]$.
5: Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$ with respect to the matrix $\bar{A}_k$.
6: Compute Z_{k+1} such that the low-rank factor product $Z_{k+1} Z_{k+1}^T$ approximates the solution of $\bar{A}_k^T X_{k+1} + X_{k+1}\bar{A}_k = -N_k N_k^T$.
7: $t_{k+1} = t_k + h$.
8: **end for**

3.2.2 Rosenbrock method of the second order

Let us now turn our attention to the second order method originally proposed in [20]. There, the method is applied to atmospheric dispersion problems describing photochemistry, advective, and turbulent diffusive transport. As explained in [15], the method can be efficiently implemented for solving autonomous RDEs as:

$$X_{k+1} = X_k + \frac{3}{2} h K_1 + \frac{1}{2} h K_2 \tag{3.12}$$

$$\bar{A}_k^T K_1 + K_1 \bar{A}_k = -F(X_k) \tag{3.13}$$

$$\bar{A}_k^T K_{21} + K_{21}\bar{A}_k = -h^2 K_1 S K_1 + \left(\frac{1}{h\gamma} + 2\right) K_1 \tag{3.14}$$

$$K_2 = K_{21} + (1 - h) K_1 \tag{3.15}$$

where $\bar{A}_k = A - SX_k - \frac{1}{2h\gamma} I$ and γ is a parameter which can be chosen as 1. Moreover, note that (3.12) and (3.15) can be computed directly as one step iteration like

$$X_{k+1} = X_k + \left(\frac{2h + h^2}{2}\right) K_1 + \frac{1}{2} h K_{21} \tag{3.16}$$

As for the linearly implicit Euler method, we want to apply the ADI iteration to solve the Lyapunov equations (3.13) and (3.14). Once again let us assume (3.10) and note that,

$$\begin{aligned} A^T Z_k Z_k^T + Z_k Z_k^T A &= A^T Z_k (Z_k^T A + Z_k^T) + Z_k (Z_k^T A + Z_k^T) - A^T Z_k Z_k^T A - Z_k Z_k^T \\ &= (A^T Z_k + Z_k)(Z_k^T A + Z_k^T) - A^T Z_k Z_k^T A - Z_k Z_k^T \\ &= (A^T Z_k + Z_k)(A^T Z_k + Z_k)^T - [\, A^T Z_k \;\; Z_k \,][\, A^T Z_k \;\; Z_k \,]^T. \end{aligned}$$

Algorithm 3.5 LRF Rosenbrock method of the second order

Require: $A \in \mathbb{R}^{n\times n}$, B, C, Z_0 satisfying (3.10), $t \in [a, b]$, and step size h.
Ensure: (Z_k, t_k) such that $X_k \approx Z_k Z_k^T$, $Z_k \in \mathbb{R}^{n\times z_i}$ with $z_i \ll n$.

1: $t_0 = a$.
2: **for** $k = 0$ to $\lceil \frac{b-a}{h} \rceil$ **do**
3: $\bar{A}_k = A - SX_k - \frac{1}{2h\gamma} I$.
4: Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$ with respect to the matrix $\bar{A}_k$.
5: $U_k = [\, C^T \;\; A^T Z_k + Z_k \,]$.
6: Compute $\tilde{T}_1^k$ such that the low-rank factor product $\tilde{T}_1^k (\tilde{T}_1^k)^T$ approximates the solution of $\bar{A}_k^T \tilde{K}_1 + \tilde{K}_1 \bar{A}_k = -U_k U_k^T$.
7: $N_k = [\, A^T Z_k \;\; Z_k \;\; Z_k (Z_k^T B) \,]$.
8: Compute $\hat{T}_1^k$ such that the low-rank factor product $\hat{T}_1^k (\hat{T}_1^k)^T$ approximates the solution of $\bar{A}_k^T \hat{K}_1 + \hat{K}_1 \bar{A}_k = -N_k N_k^T$.
9: $T_k^1 = [\, \tilde{T}_1^k \;\; i\hat{T}_1^k \,]$.
10: $\bar{N}_k = \left[\, C^T \;\; hT_k^1 ((T_k^1)^T B) \;\; i\sqrt{\frac{1}{h\gamma} + 2}\, T_k^1 \,\right]$.
11: Compute T_k^2 such that the low-rank factor product $T_k^2 (T_k^2)^T$ approximates the solution of $\bar{A}_k^T \hat{K}_{21} + \hat{K}_{21} \bar{A}_k = -\bar{N}_k \bar{N}_k^T$.
12: $Z_{k+1} = \left[\, Z_k \;\; \sqrt{\frac{2h+h^2}{2}}\, T_k^1 \;\; \sqrt{\frac{1}{2}h}\, T_k^2 \,\right]$.
13: $t_{k+1} = t_k + h$.
14: **end for**

Denoting $U_k = [\, C^T \;\; A^T Z_k + Z_k \,]$ and $N_k = [\, A^T Z_k \;\; Z_k \;\; Z_k (Z_k^T B) \,]$ then,

$$F(X_k) = U_k U_k^T - N_k N_k^T \tag{3.17}$$

thus, in order to solve (3.13) we can split the Lyapunov equation. The second equation (3.14) is solved using the factored form, for the right-hand side,

$$\bar{N}_k = \left[\, C^T \;\; hT_k^1 ((T_k^1)^T B) \;\; i\sqrt{\frac{1}{h\gamma} + 2}\, T_k^1 \,\right]$$

where $K_1 = T_k^1 (T_k^1)^T$, i.e., T_k^1 represents the low-rank factor of K_1 in step k. The method is sketched in Algorithm 3.5. There, Steps 6 and 8 can be computed simultaneously by the factored ADI iteration as a linear systems of equations to be solved in each step that have the same coefficient matrices.

Remark 3.1. The computation of Algorithm 3.5 is performed in complex arithmetics (Steps 9 and 10 computed complex factors). It is possible to keep the computation in real arithmetics such that the method works only with factors of the form $(\hat{T}_k, \tilde{T}_k)$ which approximate the solution $X_k \approx Z_0 Z_0^T + h(\hat{T}_k \hat{T}_k^T - \tilde{T}_k \tilde{T}_k^T)$. However, the number of columns increases considerably in each step and the ADI iteration slows down [38]. A remedy for this problem was recently suggested in [34], where instead of a low-rank Cholesky-like factorization, a low-rank LDL^T factorization is employed.

Remark 3.2. An adaptive code in terms of the low-rank factors can be easily implemented taking into account that the first stage of the method could be used to estimate the error and changing the step size.

3.2.3 The non-autonomous case

The ideas described above can be extended to the non-autonomous case. In this case, $\partial F/\partial t$ appears, so in order to obtain a low-rank version of the method this term, or an approximation of it, has to be represented as a low-rank matrix product combination.

If we approximate the derivatives involved in F_{t_k}, e.g., using central differences as:

$$\dot{Q}_k := \frac{Q_{k+1} - Q_{k-1}}{2h}, \qquad \dot{A}_k := \frac{A_{k+1} - A_{k-1}}{2h}, \qquad \dot{S}_k := \frac{S_{k+1} - S_{k-1}}{2h}$$

(note that, in the context of RDEs arising in optimal control the matrix $Q(t)$ is generally constant, it represents the output matrix), then F_{t_k} can be approximated by

$$\begin{aligned} F_{t_k} \approx \frac{1}{2h}\Big[&(Q_{k+1} - Q_{k-1}) + 2hA_k^T F(X_k) + (A_{k+1}^T - A_{k-1}^T)X_k + 2hF(X_k)A_k + X_k(A_{k+1} \\ &- A_{k-1}) - 2hF(X_k)S_kX_k - X_k(S_{k+1} - S_{k-1})X_k - 2hX_kS_kF(X_k)\Big]. \end{aligned} \tag{3.18}$$

By (3.17) we know that $F(X_k)$ can be expressed as a combination of low-rank factor matrix products, then by re-arranging terms we can obtain a low-rank matrix representation of (3.18). Finally, we point out that although we have focused on two particular Rosenbrock methods the ideas can be applied to a general s-stage Rosenbrock method or to an embedded method described in [15].

4 Numerical results

4.1 Diffusion problem

We consider a diffusion problem on a square domain (see Fig. 2). The equations that describe the model have the form

$$\begin{aligned} \varrho x_t(\xi, t) &= \Delta x(\xi, t) && \text{in } \Omega \times (0, T) \\ \partial_\nu x(\xi, t) &= b(\xi)u(t) - x(\xi, t) && \text{on } \Gamma_c \\ \partial_\nu x(\xi, t) &= -x(\xi, t) && \text{on } \partial\Omega\backslash\Gamma_c \\ x(\xi, 0) &= 1 && \text{in } \Omega \end{aligned} \tag{4.1}$$

where $x(\xi, t)$ represent the state at time t in the point ξ, $b(\xi) = 4(1-\xi)\xi$ for $\xi \in \Gamma_c$ and 0 otherwise. Thus, for all $t \in \mathbb{R}^+$, we have $u(t) \in \mathbb{R}$. The problem parameters chosen can be found in Table 2. There, n is the dimension, Q, R, G are the operators from the finite-dimensional LQR problem, h is the step size (or the initial step size) and T_f is the time horizon.

We use the finite element software package FEINS [46] to generate the data for different mesh sizes, $n = 289, 1089, 4225, 16641, 66049$. In Fig. 1, the initial and controlled (at $0.01s$) states are plotted. We applied the BDF method up to order three with variable step size for $n = 289, 1089$; and for the refined meshes $n = 4225, 16641, 66049$, the linearly implicit Euler method was applied.

In Table 1 the cost functional values, computed by (2.10), are shown. These values are plotted over the mesh size in Fig. 3.

We can visualize the convergence result from Section 2.1 as the cost functional values are computed by (2.10). But, we have not sufficiently many refinements at hand to provide a reasonable estimate of the convergence rate based on the the experimental order of convergence (EOC).

n	Boundary	n	Diffusion
371	9.1375e+7	289	1.2881034e-1
1357	5.0823e+7	1089	1.1592931e-1
5177	4.0613e+7	4225	1.1478002e-1
20209	3.9508e+7	16641	1.1476854e-1
79841	3.9506e+7	66049	1.1476848e-1

Tab. 1: Cost functional values for the boundary control problem and the diffusion problem.

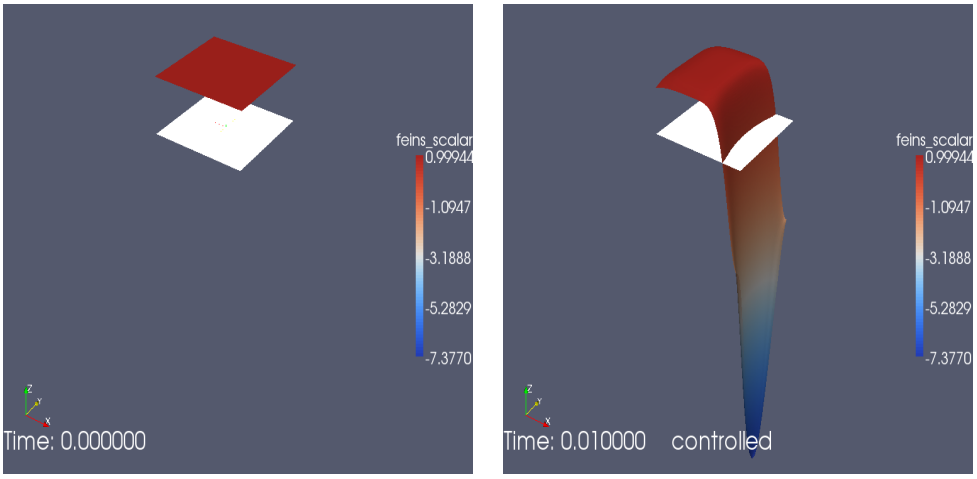

Fig. 1: Diffusion problem on a square domain: initial state (left) and controlled state at time 0.01 (right).

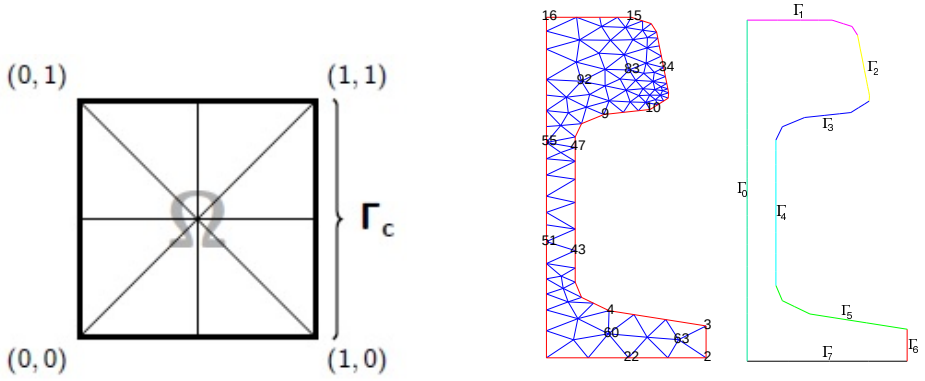

Fig. 2: Domain diffusion problem (left) and boundary control problem (right).

4.2 Boundary control problem

Let us consider a boundary control problem for the heat equation. This problem arises in a rolling mill when different steps in the production process require different temperatures of the raw material. An infinitely long steel profile is assumed so that a 2-dimensional heat diffusion process is considered. Exploiting the symmetry of the workpiece, an artificial boundary Γ_0 is introduced on the symmetry axis (see Fig. 2). A linearized version of the model has the form

$$\begin{aligned} c\varrho x_t(\xi,t) &= \lambda\Delta x(\xi,t) && \text{in } \Omega\times(0,T)\\ -\lambda\partial_\nu x(\xi,t) &= g_i(t,x,u) && \text{on } \Gamma_i,\quad i=0,\dots,7\\ x(\xi,0) &= x_0(\xi) && \text{in } \Omega \end{aligned} \tag{4.2}$$

Test	n	Q	R	G	T_f	h
1	289	I	I	0	1	1e-3
2	1089	I	I	0	1	1e-3
3	4225	I	I	0	1	1e-3
4	16641	I	I	0	1	1e-3
5	66049	I	I	0	1	1e-3

Tab. 2: Control parameters for the diffusion problem.

Test	n	Q	R	G	T_f	h
1	371	I	I	0	20	1e-2
2	1357	I	I	0	20	1e-2
3	5177	I	I	0	20	1e-2
4	20209	I	I	0	20	1e-2
5	79841	I	I	0	20	1e-2

Tab. 3: Parameters for the boundary control problem.

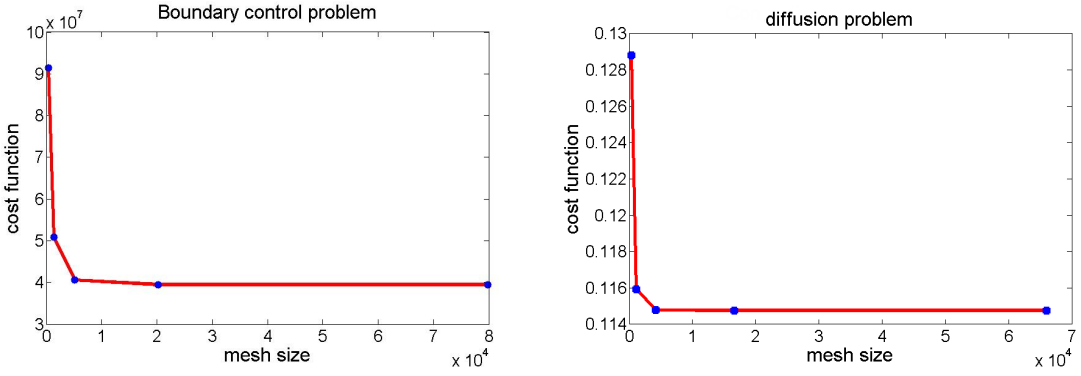

Fig. 3: Cost functional values for different mesh sizes for the boundary control problem (left) and the diffusion problem (right)

where $x(\xi, t)$ represent the temperature at time t in point ξ, g_i includes temperature differences between cooling fluid and profile surface, intensity parameters for the cooling nozzles and heat transfer coefficients modeling the heat transfer to cooling fluid [12, 44, 48].

We applied the BDF method up to order three with variable step size for $n = 371, 1357$; and for the refined meshes $n = 5177, 20209, 79841$, the Rosenbrock method of order one was applied.

The problem parameters chosen can be found in Table 3. There, n is the dimension, Q, R, G are the operators from the finite-dimensional LQR problem, h is the temporal step size (or the initial step size) and T_f is the time horizon.

In Table 1 the cost functional values, computed by (2.10), are shown. These values are plotted over the mesh size in Fig. 3. We also computed the EOC based on these numbers, like in a similar numerical experiment performed in [16]. The obtained numbers are much better than expected from the mesh refinement (bisection) and do not settle to a value, but keep increasing. As we only have five mesh instances reported here, we refrain from reporting the numbers here as they give no indication of the actual convergence rate of the cost functional.

4.3 Convection–diffusion problem

We consider a finite differences semi-discretized heat equation with convection on the unit square with homogeneous first kind boundary conditions,

$$x_t(\xi, t) - v \cdot \nabla x(\xi, t) - \Delta x(\xi, t) = f(\xi)u(t) \quad \text{in } \Omega \times (0, T) \tag{4.3}$$

where $x(\xi, t)$ represent the state at time t in the point ξ, $\Omega := (0, 1) \times (0, 1)$ and the vector v is chosen as $v = [10\ \ 100]^T$. This problem is used in LyaPack demonstration scripts [41].

We applied the BDF method up to order three with variable step size for $n = 1600, 2500$ and for the refined meshes $n = 3600, 4900, 6400, 10000, 40000$ the Rosenbrock method of order one was applied. In Table 4 the cost functional values, computed by (2.10), are shown. These values are plotted over the mesh size in Fig. 4. Once again we can visualize the convergence result from Section 2.1.

n	Convection–diffusion
1600	1.43730e+5
2500	1.42300e+5
3600	1.40900e+5
4900	1.28090e+5
6400	0.62760e+5
10000	0.60820e+5
40000	0.60810e+5

Tab. 4: Cost functional values for the convection–diffusion problem.

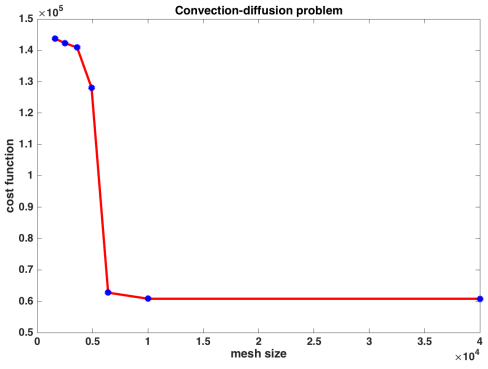

Fig. 4: Cost functional values for different mesh sizes for the convection–diffusion problem.

5 Conclusions

The numerical treatment of linear quadratic regulator problems for parabolic partial differential equations on a finite-time horizon requires solving large-scale RDEs resulting from semi-discretization. In order to provide an approximation framework for the computation of the finite-dimensional Riccati equations, we have shown the convergence of the infinite-dimensional Riccati operators to the finite-dimensional ones for the (non)autonomous case. Typically the coefficient matrices of the resulting RDE there have a given structure, e.g., sparse, symmetric, low-rank. We develop efficient numerical methods capable of exploiting this structure based on matrix-valued versions of the BDF and Rosenbrock methods. The implementation uses a low-rank ADI iteration for solving the Lyapunov equations arising in the methods. The crucial question of suitable step-size and order selection strategies is also addressed in terms of the low-rank factors of the solution. The numerical experiments confirm the good performance of the proposed methods and show their potential for being used in large-scale problems.

Acknowledgment: We thank Jens Saak for providing the first example. We also would like to thank the referees for their valuable comments. They greatly helped to improve this manuscript.

References

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

[2] A. C. Antoulas, D. C. Sorensen, and Y. Zhou, On the decay rate of Hankel singular values and related issues, *Syst. Contr. Lett.* **46** (2000), No. 5, 323–342.

[3] E. Arias, V. Hernández, J. Ibanes, and J. Peinado, A family of BDF algorithms for solving Differential Matrix Riccati Equations using adaptive techniques, *Procedia Computer Science* **1** (2010), No. 1, 2569–2577.

[4] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, PA, SIAM, Philadelphia, 1998.

[5] H. T. Banks and K. Kunisch, The linear regulator problem for parabolic systems, *SIAM J. Cont. Optim.* **22** (1984), 684–698.

[6] P. Benner, Solving large-scale control problems, *IEEE Control Systems Magazine* **14** (2004), No. 1, 44–59.

[7] P. Benner and S. Hein, Model predictive control for nonlinear parabolic differential equations based on a linear quadratic Gaussian design, In: *Proc. in Applied Mathematics and Mechanics*, 9, pp. 613–614, 2009.
[8] P. Benner and S. Hein, *Model Predictive Control Based on an LQG Design for Time-Varying Linearizations*, Chemnitz Scientific Computing Preprints, Report No. 09-07, TU Chemnitz (Germany), 2010.
[9] P. Benner and S. Hein, MPC/LQG for infinite-dimensional systems using time-invariant linearizations, In: *System Modeling and Optimization. 25th IFIP TC 7 Conference, Berlin, Germany, September 12-16*, 391, pp. 217–224, IFIP AICT, 2011.
[10] P. Benner, P. Kürschner, and J. Saak, Efficient handling of complex shift parameters in the low-rank cholesky factor ADI method, *Numerical Algorithms* **62** (2012), No. 2, 225–251.
[11] P. Benner, J. R. Li, and T. Penzl, Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems, *Numer. Lin. Alg. Appl.* **15** (2008), No. 9, 755–777.
[12] P. Benner, V. Mehrmann, and D. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems*, Lecture Notes in Computational Science and Engineering 45, Springer-Verlag, Berlin/Heidelberg, Germany, 2005.
[13] P. Benner and H. Mena, BDF methods for large-scale differential Riccati equations, In: *Proc. of Mathematical Theory of Network and Systems, MTNS 2004* (Eds. B. De Moor, B. Motmans, J. Willems, P. Van Dooren, and V. Blondel), 2004.
[14] P. Benner and H. Mena, *Numerical Solution of the Infinite-Dimensional LQR-Problem and the Associated Differential Riccati Equations*, Max Planck Institute Magdeburg, Preprint No. MPIMD/12-13, August 2012, Available from http://www.mpi-magdeburg.mpg.de/preprints/.
[15] P. Benner and H. Mena, Rosenbrock methods for solving differential Riccati equations, *IEEE Trans. Automat. Control* **58** (2013), No. 11, 2950–2957.
[16] P. Benner and J. Saak, Suboptimality estimates for the semi-discretized LQR probelm for parabolic PDEs, In: *Proc. in Applied Mathematics and Mechanics*, 10, pp. 591–592, 2010.
[17] P. Benner and J. Saak, Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey, *GAMM Mitteilungen* **36** (2013), No. 1, 32–52.
[18] A. Bensoussan, G. Da Prato, M. C. Delfour, and S. K. Mitter, *Representation and Control of Infinite Dimensional Systems, Volume I*, Systems & Control: Foundations & Applications, Birkäuser, Boston–Basel–Berlin, 1992.
[19] A. Bensoussan, G. Da Prato, M. C. Delfour, and S. K. Mitter, *Representation and Control of Infinite Dimensional Systems, Volume II*, Systems & Control: Foundations & Applications, Birkäuser, Boston–Basel–Berlin, 1992.
[20] J. G. Blom, W. Hundsdorfer, E. J. Spee, and J. G. Verwer, A second order Rosenbrock method applied to photochemical dispersion problems, *SIAM J. Sci. Comput.* **20** (1999), No. 4, 1456–1480.
[21] F. Bornemann and P. Deuflhard, *Scientific Computing with Ordinary Differential Equations*, Text in Applied Mathematics 42, Springer, New York, 2002.
[22] C. Choi and A. J. Laub, Efficient matrix-valued algorithms for solving stiff Riccati differential equations, *IEEE Trans. Automat. Control* **35** (1990), 770–776.
[23] R. F. Curtain and A. J. Pritchard, Infinite-dimensional Riccati equation for systems defined by evolution operators, *SIAM J. Cont. Optim.* **14** (1976), 951–983.
[24] R. F. Curtain and H. Zwart, *An Introduction to Infinite-Dimensional Linear System Theory*, Texts in Applied Mathematics, Springer-Verlag, New York, 1995.
[25] L. Dieci, Numerical integration of the differential Riccati equation and some related issues, *SIAM J. Numer. Anal.* **29** (1992), No. 3, 781–815.
[26] E. Eich, *Projizierende Mehrschrittverfahren zur numerischen Lösung von Bewegungsgleichungen technischer Mehrkörpersysteme mit Zwangsbedingungen und Unstetigkeiten*, Ph.D. thesis, University of Augsburg, 1991.
[27] J. S. Gibson, The Riccati integral equation for optimal control problems in Hilbert spaces, *SIAM J. Cont. Optim.* **17** (1979), No. 4, 537–565.
[28] L. Grasedyck, Existence of a low rank or H-matrix approximant to the solution of a Sylvester equation, *Numer. Lin. Alg. Appl.* **11** (2004), 371–389.
[29] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II-Stiff and Differerntial Algebraic Problems*, Springer Series in Computational Mathematics, Springer-Verlag, New York, 2000.
[30] S. Hein, *MPC/LQG-Based Optimal Control of Nonlinear Parabolic PDEs*, Ph.D. thesis, TU Chemnitz, February 2010, available from http://nbn-resolving.de/urn:nbn:de:bsz:ch1-201000134.
[31] K. Ito and K. Kunisch, Receding horizon optimal control for infinite dimensional systems, *ESAIM Control Optim. Calc. Var.* **8** (2002), 741–760.
[32] K. Ito and K. Kunisch, Receding horizon control with incomplete observations, *SIAM J. Cont. Optim.* **45** (2006), No. 1, 207–225.
[33] M. Kroller and K. Kunisch, Convergence rates for the feedback operators arising in the linear quadratic regulator problem governed by parabolic equations, *SIAM J. Numer. Anal.* **28** (1991), No. 5, 1350–1385.
[34] N. Lang, H. Mena, and J. Saak, On the benefits of the LDL factorization for large-scale differential matrix equation solvers, *Lin. Alg. Appl.* **480** (2015), 44–71.
[35] I. Lasiecka and R. Triggiani, *Differential and Algebraic Riccati Equations with Application to Boundary/Point Control Problems: Continuous Theory and Approximation Theory*, Lecture Notes in Control and Information Sciences 164, Springer, Berlin, 1991.

[36] I. Lasiecka and R. Triggiani, *Control Theory for Partial Differential Equations: Continuous and Approximation Theories I. Abstract Parabolic Systems*, Cambridge University Press, Cambridge, UK, 2000.
[37] J. R. Li and J. White, Low rank solution of Lyapunov equations, *SIAM J. Matrix Anal. Appl.* **24** (2002), No. 1, 260–280.
[38] 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.
[39] T. Penzl, *Numerische Lösung großer Lyapunov-Gleichungen*, Ph.D. thesis, Technische Universität Chemnitz, 1998.
[40] T. Penzl, Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case, *Sys. Control Lett.* **40** (2000), 139–144.
[41] T. Penzl, *Lyapack Users Guide*, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, Report No. SFB393/00-33, 09107 Chemnitz, Germany, 2000, Available from http://www.tu-chemnitz.de/sfb393/sfb00pr.html.
[42] I. R. Petersen, V. A. Ugrinovskii, and A. V. Savkin, *Robust Control Design Using H^∞ Methods*, Springer-Verlag, London, UK, 2000.
[43] A. J. Pritchard and D. Salamon, The linear quadratic control problem for infinite dimensional systems with unbounded input and output operators, *SIAM J. Cont. Optim.* **25** (1987), 121–144.
[44] J. Saak, *Effiziente numerische Lösung eines Optimalsteuerungsproblems für die Abkühlung von Stahlprofilen*, Diplomarbeit, Fachbereich 3/Mathematik und Informatik, Universität Bremen, D-28334 Bremen, 2003.
[45] J. Sabino, *Solution of Large-Scale Lyapunov Equations via the Block Modified Smith Method*, Ph.D. thesis, Rice University, Houston, Texas, 2007.
[46] R. Schneider, FEINS: Finite element solver for shape optimization with adjoint equations, In: *Progress in Industrial Mathematics at ECMI 2010*, Mathematics in Industry, Springer, 2011, Software available at http://www.feins.org/.
[47] V. Simoncini, A new iterative method for solving large-scale Lyapunov matrix equations, *SIAM J. Sci. Comput.* **29** (2007), No. 3, 1268–1288.
[48] F. Tröltzsch and A. Unger, Fast solution of optimal control problems in the selective cooling of steel, *Z. Angew. Math. Mech.* **81** (2001), 447–456.
[49] X. Wu, B. Jacob, and H. Elbern, *Optimal Actuator and Observation Location for Time-Varying Systems on a Finite-Time Horizon*, arXiv:1503.09031, March 2015.
[50] J. Zabczyk, Remarks on the algebraic Riccati equation, *Appl. Math. Optim.* **2** (1976), 251–258.