Hermann Mena Peter Benner

Numerical solution of the Infinite-Dimensional LQR-Problem and the associated Differential Riccati Equations

Max Planck Institute Magdeburg Preprints

MPIMD/12-13 **August 8, 2012**

Impressum:

Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Publisher:
Max Planck Institute for Dynamics of Complex Technical Systems

Address:
Max Planck Institute for Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg

www.mpi-magdeburg.mpg.de/preprints

NUMERICAL SOLUTION OF THE INFINITE-DIMENSIONAL LQR-PROBLEM AND THE ASSOCIATED DIFFERENTIAL RICCATI EQUATIONS

PETER BENNER* AND HERMANN MENA†

Abstract. The numerical analysis of linear quadratic regulator design problems for parabolic partial differential equations requires solving large-scale Riccati equations. In the finite time horizon case, the differential Riccati equation (DRE) arises. Typically, the coefficient matrices of the resulting DRE have a given structure,e.g., sparse, symmetric or low rank. Moreover, in most control problems, fast and slow modes are present. This implies that the associated DRE will be fairly stiff. Therefore, implicit schemes have to be used to solve such DREs numerically. In this paper we derive efficient numerical methods for solving DREs 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 particularly suitable for large-scale problems by working only on low-rank factors of the solutions. Step size and order control strategies can also be implemented based only on information contained in the solution factors. Finally, we briefly show that within a Galerkin projection framework the solutions of the finite-dimensional DREs converge in the strong operator topology to the solutions of the infinite-dimensional DREs. The performance of each of these methods is tested in numerical experiments.

Key words. DRE, large-scale, Rosenbrock, BDF methods

1. Introduction. The differential Riccati equation (DRE) 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, 28, 37]). In the literature, there is a large variety of approaches to compute the solution of the DRE, e.g., [16, 20]), 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 DREs arising in optimal control problems for parabolic partial differential equations. Hence, let consider nonlinear parabolic diffusion-convection and diffusion-reaction systems of the form

$$\frac{\partial \mathbf{x}}{\partial t} + \nabla \cdot (\mathbf{c}(\mathbf{x}) - \mathbf{k}(\nabla \mathbf{x})) + \mathbf{q}(\mathbf{x}) = \mathbf{B}\mathbf{u}(t), \qquad t \in [0, T_f], \tag{1.1}$$

in $\Omega \subset \mathbb{R}^d, d = 1, 2, 3$, with appropriate initial and boundary conditions. The equation can be split into the convective term $\mathbf{c}$, the diffusive part $\mathbf{k}$ and the uncontrolled reaction given by $\mathbf{q}$. The state $\mathbf{x}$ of the system depends on $\xi \in \Omega$ and the time $t \in [0, T_f]$ and is denoted by $\mathbf{x}(\xi, t)$. Moreover, we will focus on applications where the control $\mathbf{u}(t)$ is assumed to depend only on the time $t \in [0, T_f]$, while the linear operator $\mathbf{B}$ may depend on $\xi \in \Omega$.
If (1.1) is linear, then a variational formulation leads to an abstract Cauchy problem for a linear evolution equation of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \qquad \mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{H}, \tag{1.2}$$

for linear operators $\mathbf{A} : \mathrm{dom}(\mathbf{A}) \subset \mathcal{H} \to \mathcal{H}$, $\mathbf{B} : \mathcal{U} \to \mathcal{H}$, $\mathbf{C} : \mathcal{H} \to \mathcal{Y}$; where the state space $\mathcal{H}$, the observation space $\mathcal{Y}$, and the control space $\mathcal{U}$ are assumed to be

*Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany benner@mpi-magdeburg.mpg.de

†Departamento de Matemática, Escuela Politécnica Nacional, Ladrón de Guevara E11-253, Quito-Ecuador, hermann.mena@epn.edu.ec

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., $\mathbf{y} = \mathbf{C}\mathbf{x}$.
If (1.1) is nonlinear, model predictive control technics can be applied [25, 26]. There the equation is linearized at certain working points or around reference trajectories and linear problems for equations as in (1.2) have to be solved on subintervals of $[0, T_f]$.

Let us consider that the cost functional $\hat{J}(\mathbf{x}, \mathbf{u})$ is given in a quadratic form, i.e.,

$$J(\mathbf{u}) = \frac{1}{2} \int_0^{T_f} \langle \mathbf{x}, \mathbf{Q}\mathbf{x} \rangle_{\mathcal{H}} + \langle \mathbf{u}, \mathbf{R}\mathbf{u} \rangle_{\mathcal{U}} \, dt + \langle \mathbf{x}_{T_f}, \mathbf{G}\mathbf{x}_{T_f} \rangle_{\mathcal{H}}, \tag{1.3}$$

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 $\mathbf{x}(., T_f)$ and $T_f < \infty$. As often 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 that: $\mathbf{A}$ is the infinitesimal generator of a strongly continuous semigroup $T(t)$, $\mathbf{B}, \mathbf{C}$ are linear bounded operators and for every initial value there exists an admissible control $\mathbf{u} \in L^2(0, \infty; \mathcal{U})$ hold, then the solution of the abstract LQR problem can be obtained analogously to the finite-dimensional case [18, 22, 31, 43] as a *feedback control*

$$\mathbf{u}_*(t) = -\mathbf{R}^{-1}\mathbf{B}^*\mathbf{X}_\infty(t)\mathbf{x}_*(t), \tag{1.4}$$

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

$$\dot{\mathbf{X}}(t) = -(\mathbf{C}^*\mathbf{Q}\mathbf{C} + \mathbf{A}^*\mathbf{X}(t) + \mathbf{X}(t)\mathbf{A} - \mathbf{X}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{X}(t)) \tag{1.5}$$

with terminal condition $\mathbf{X}_{T_f} = \mathbf{G}$. Most of the required conditions, particularly the restrictive assumption that $\mathbf{B}$ is bounded, can be weakened [30, 31, 39].
In order to solve the infinite-dimensional LQR problem numerically, we use a Galerkin projection of the variational formulation of (1.1) 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 DREs resulting from the semi-discretization. Typically, the coefficient matrices of the DRE have a given structure (e.g. sparse, symmetric, or low rank). Moreover, we expect to treat stiff DREs, so we will focus on methods that can efficiently deal with stiffness. Here, 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 can also be implemented based only on information contained in the solution factors.
The task of solving large-scale DREs has also become an important issue in nonlinear optimal control problems of tracking type and stabilization. Linear-quadratic Gaussian (LQG) design on short time intervals is the main computational ingredient in recently proposed receding horizon (RHC) and model predictive control (MPC) approaches. There, linear problems have to be solved on the time frames [25, 26, 27].
This paper is organized as follows: in the next section we present an approximation framework for computation of Riccati operators. After that, efficient matrix valued

algorithms of the BDF and Rosenbrock methods for large-scale DREs are proposed in Section 3. The crucial question of suitable stepsize and order selection strategies is also addressed. In Section 4 numerical examples for linear and nonlinear control problems show the performance of the methods. Some conclusions and outlook are summarized in 5.

2. Theoretical aspects. The linear-quadratic control problem for infinite-dimensional systems has been deeply studied [11, 12, 31, 32]. Particularly, approximation schemes for Riccati equations in infinite-dimensional spaces have been proposed in the last years. Chronologically, the first reference is Gibson [22], 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 [29, 31, 32].
For the finite-time horizon case, we propose an approximation scheme in terms of differential Riccati equations. The finite-dimensional approximating problems are each defined on a subspace of the state space of the original problem. The proof follows from the abstract theory developed by Gibson [22], and from the ideas for the infinite-time horizon case presented in [5]. The result that is shown here can be seen as a corollary of the ones proposed by Gibson [22] and Banks/Kunisch [5]. However, for completeness we included it.
If we semi-discretize an infinite-dimensional linear-quadratic regulator (LQR) problem in space, we obtain a finite-dimensional LQR problem. In this section, for the finite-time horizon case, we study the convergence of the finite-dimensional Riccati operators (i.e., the operators related to a matrix DRE) to the infinite-dimensional ones. For simplicity we consider first 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 $\mathbf{A}$: $\mathrm{dom}(\mathbf{A}) \subset \mathcal{H} \to \mathcal{H}$ 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 (1.2) and the cost functional (1.3). We assume that (1.2) has a unique solution [31, 19]. 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 $\tilde{\mathbf{Q}} \geq 0$, $\mathbf{R} > 0$, $\mathbf{G} \geq 0$ and $\mathbf{x}_{T_f}$ denotes $\mathbf{x}(., T_f)$.
We will say that a function $u \in L^2(0, T_f; \mathcal{U})$ is an admissible control for the initial state $\mathbf{x}_0 \in \mathcal{H}$ if $J(x_0, \mathbf{u})$ in (1.3) is finite. Let us consider the operator differential Riccati equation. We define a solution of (1.5) in the interval $[0, T_f]$ as an operator $\mathbf{\Pi}(t)$ such that $\mathbf{\Pi}(T_f) = \mathbf{G}$ and for all φ, $\psi \in \mathrm{dom(A)}$, $\langle \varphi, \mathbf{\Pi}(.)\psi \rangle$ is differentiable in $[0, T_f]$ and satisfies the equation,

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

as is defined in [11, Def. 2.1, pp. 142]. Note that any solution of (1.5) is self-adjoint, and that $\mathbf{\Pi}(.)$ is nonnegative if $\mathbf{G}$ is. In order to solve numerically the operator differential Riccati 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 strongly 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, $Q^N \in \mathcal{L}(\mathcal{H}^N)$, $G^N \geq 0$, we consider the family of linear-quadratic regulator problems on $\mathcal{H}^N$:

Minimize:

$$\begin{aligned} J(x_0^N, \mathbf{u}) &:= \int_0^{T_f} \langle x^N, Q^N x^N \rangle_{\mathcal{H}^N} + \langle \mathbf{u}, \mathbf{R}\mathbf{u} \rangle_{\mathcal{U}} dt \\ &\quad + \langle x_{T_f}^N, G^N x_{T_f}^N \rangle_{\mathcal{H}}^N. \end{aligned} \tag{$\mathcal{R}^N$}$$

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 $Q^N \geq 0$, $\mathbf{R} > 0$, then the optimal control for ($\mathcal{R}^N$) is given in feedback form by

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

where $\Pi^N(t) \in \mathcal{L}(\mathcal{H}^N)$ is the unique nonnegative self-adjoint solution of the differential Riccati 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 $\mathbf{u}(t) = u_*(t)^N$, [1]. Let us now consider a related family of regulator problems, in which the operators are defined on the whole space,

Minimize:

$$\begin{aligned} J(x_0^N, \mathbf{u}) &:= \int_0^{T_f} \langle x^N, \bar{\mathbf{Q}}^N x^N \rangle_{\mathcal{H}} + \langle \mathbf{u}, \mathbf{R}\mathbf{u} \rangle_{\mathcal{U}} dt \\ &\quad + \langle x_{T_f}^N, \bar{\mathbf{G}}^N x_{T_f}^N \rangle_{\mathcal{H}} \end{aligned} \tag{$\bar{\mathcal{R}}^N$}$$

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

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

Note that

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

In fact, if in (2.2) we replace Q^N, A^N, G^N by $Q^N P^N$, $A^N P^N$, $G^N P^N$, 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.

2.1. Convergence statement. The convergence result of this section is essentially contained in [22]. 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 [22] 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 $N \to \infty$

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

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

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

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 $\mathbf{\Pi}(t)$ be the unique element of $\mathcal{B}_\infty(0,T_f;\mathcal{H},\mathcal{H})$, is the set of all such functions essentially bounded on $[0, T_f]$, which satisfies the first Riccati integral equation. By calculations in [22, pp. 544-546], $\mathbf{\Pi}(t)$ is also the unique solution of the Riccati integral equation of Curtain and Pritchard [17]. Moreover, $\mathbf{\Pi}(t)$ uniquely satisfies the infinite-dimensional differential Riccati equation (2.1). Let $\bar{\mathbf{\Pi}}^N(t)$ 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 [22, 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 $(\mathcal{H}, \|.\|)$, $(\mathcal{H}^N, \|.\|_N)$ 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:

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

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.,

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

The proof of this result follows very close to the one of Theorem 2.1 once an analogue to [22, Theorem 5.1, p. 560], which permits $\mathcal{H}^N \nsubseteq \mathcal{H}$, has been proven. Note that [22, Theorem 5.1, p. 560] relies directly on [22, Lemma 5.1, p. 560]. This Lemma can be modified as:

LEMMA 2.2. *Let X be a Banach space, let $\{X^N\}_{N\geq 2}$ be a sequence of Banach spaces and let $P^N : X \to X^N$ be bounded linear operators satisfying (H')(0). Let Ω be a compact subset of $\mathbb{R}^n$ and let $A(\cdot) : \Omega \to \mathcal{L}(X)$, and for $N \geq 2$, 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 $\xi \in \Omega$, note that

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

then, by the hypotheses assumed the lemma holds □.

The repeated application of Lemma 2.2, and Lemma 5.1 [22, p. 560] allows also to prove an analogous result for the non-autonomous case which permits $\mathcal{H}^N \nsubseteq \mathcal{H}$. This version of the theorem could be very useful for developing certain types of approximation schemes, e.g., finite differences or spectral methods.

2.1.1. Non-autonomous case. The approximation results presented above can be extended to the non-autonomous case, i.e., the case, in which partial differential equations with time-varying coefficients are considered. These can be particularly useful when 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 DREs, in which the coefficient matrices are time dependent, [25, 27].

We consider now partial differential equations with time-varying coefficients. Then, 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

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

where $0 \leq s \leq t \leq T_f < \infty$, $U(.,.)$ is a strong evolution operator on $\mathcal{H}$, $u \in L^2(0,T_f;\mathcal{U})$, $x_0 \in \mathcal{H}$, and $\mathbf{B} \in \mathcal{B}_\infty(0,T_f;\mathcal{H},\mathcal{H})$.
Note that (2.7) can be differentiated using

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

where $\mathbf{A}(.)$ is the generator of $U(.,.)$ and $\mathcal{D}_\mathbf{A}$ is the domain, in which $U(.,.)$ is a mild evolution family. We use the integral form of (2.7) in our presentation to closely follow [17, 22]. We consider the cost functional

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

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

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

Then, the quadratic cost problem is:

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

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 $v \in \mathcal{U}$,

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

In addition we require

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

to be uniformly bounded in N, t, and s and require a constant m such that for each N, $Q^N(t) \ge m > 0$ 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 [22]. 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 [22] 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 2.3. *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 $\mathbf{\Pi}(\cdot)$. Then we have*

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

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

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

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 a subspaces of the original state space similar to ($\mathcal{R}^N$). Let us denote these problems as ($\mathcal{NAR}^{\mathcal{N}}$). If we consider a related family of control problems ($\overline{\mathcal{NAR}^{\mathcal{N}}}$) which, are defined in the whole space analogous to ($\bar{\mathcal{R}}^N$), and assuming similar arguments on $\mathbf{\Pi}(\mathbf{t})$ to the ones in the proof of Theorem 2.1, the proof of Theorem 2.3 follows directly from [22, Theorem 5.1, p. 560].□
Like in the autonomous case, it is possible to prove an analogue to Theorem 2.3 without the requirement $\mathcal{H}^N \subseteq \mathcal{H}$. Let us assume that $(\mathcal{H}, \|.\|)$, $(\mathcal{H}^N, \|.\|_N)$ 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 $\left\|P^N\phi\right\|_N \to \|\phi\|$ for all $\phi \in \mathcal{H}$.

(i) There exist M, ω such that $\left\|U^N(t,s)\right\|_N \leq Me^{\omega(t-s)}$, $t \geq s$, for all N and for each $\phi \in \mathcal{H}$, $\left\|U(t,s)^N P^N\phi - P^N U(t,s)\phi\right\|_N \to 0$ as $N \to \infty$, uniformly on any bounded subinterval of $[0, T_f]$.

(ii) For all $\phi \in \mathcal{H}$ it holds $\left\|U^{N*}(t,s)P^N\phi - P^N U^*(t,s)\phi\right\|_N \to 0$ 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 $\left\|B^{N*}P^N\varphi - \mathbf{B}^*\varphi\right\|_U \to 0$. (GN')

(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 2.3, 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 2.2 and Lemma 5.1 [22, p. 560].

REMARK 2.4. *The results proposed in this subsection will be 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 DREs, in which the coefficient matrices are time dependent.*

We have seen that solving infinite-dimensional LQR problems numerically requires the solution of finite-dimensional DREs obtained from appropriate discretization schemes satisfying assumptions (H), (H') or (G'), (GN').

3. Numerical methods for large scale DREs. Note that making a simply change of variables we can solve DRE with terminal condition forward in time and afterwards recover the original solution. Then, let us consider time-varying symmetric DREs 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]. Moreover, in most control problems, fast and slow modes are present. This implies that the associated DRE will be fairly stiff, which in turn demands for implicit methods to solve such DREs numerically. Therefore, we will focus here on the stiff case.

3.1. BDF methods for large-scale DREs. In the following we briefly describe the BDF method for DREs in matrix-valued form similar to [16] and discuss an efficient implementation for large-scale problems. Parts of this section are contained

p	β	α_0	α_1	α_2	α_3	α_4
1	1	1				
2	$\frac{2}{3}$	$\frac{4}{3}$	$-\frac{1}{3}$			
3	$\frac{6}{11}$	$\frac{18}{11}$	$-\frac{9}{11}$	$\frac{2}{11}$		
4	$\frac{12}{25}$	$\frac{48}{25}$	$-\frac{36}{25}$	$\frac{16}{25}$	$-\frac{3}{25}$	
5	$\frac{60}{137}$	$\frac{300}{137}$	$-\frac{300}{137}$	$\frac{200}{137}$	$-\frac{75}{137}$	$-\frac{12}{137}$

TABLE 3.1
coefficients of the BDF p-step methods for $p < 6$.

in [8]. 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 DRE (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, given in Table 3.1, [3].

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

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

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

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

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, see [2, 36], i.e., it can be well approximated by a *low rank factor* (LRF) 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

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

where

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

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 DRE solver for large-scale problems.
In our numerical implementation, we benefit from recent algorithmic progress in solving large-scale AREs resulting from semi-discretized control problems [6]. 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. The implementation of the BDF methods for DREs is sketched in Algorithm 3.1.

REMARK 3.1. *It should be noted that for $p \geq 2$, some of the α_j are negative. This can be treated using complex arithmetic and replacing all transposes in (3.5) by conjugate complex transposes, but in general it will be more efficient to split the constant term into*

$$\hat{C}_{k+1}^T\hat{C}_{k+1} - \tilde{C}_{k+1}^T\tilde{C}_{k+1}$$

where $\hat{C}_{k+1}$ only contains the factors corresponding to positive α_j and $\tilde{C}_{k+1}$ the factors corresponding to negative α_j. Therefore, we split the Lyapunov equation into the two equations

$$A_\ell^T\hat{P}_{\ell+1} + \hat{P}_{\ell+1}A_\ell = -\hat{C}^T\hat{C} - P_\ell BB^T P_\ell,$$
$$A_\ell^T\tilde{P}_{\ell+1} + \tilde{P}_{\ell+1}A_\ell = -\tilde{C}^T\tilde{C}.$$

Then the solution of the original equation in step ℓ is computed as $\hat{P}_{\ell+1} - \tilde{P}_{\ell+1}$. The two Lyapunov equations can be solved simultaneously as a linear systems of equations to be solved in each step that have the same coefficient matrices.

3.1.1. Adaptive step size and order control. In most applications, varying the step size and order is crucial for the efficient performance of a discretization method. We start forming estimates of the error which we expect would be incurred on the next step and choosing the next order so that the step size at that order is the largest possible.
The local truncation error for the BDF methods can be written as in [21]:

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

where

$$\omega_k(t) = \prod_{i=0}^{p}(t - t_{k-i}),$$

and

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

Algorithm 3.1 LRF BDF method of order p

Require: $A(t)$, $S(t)$, $Q(t)$, $\in \mathbb{R}^{n\times n}$ smooth matrix-valued functions satisfying (3.4), $t \in [a, b]$, and h step size.
Ensure: (Z_k, t_k) such that $X_k \approx Z_k Z_k^T$.

$t_0 = a$.
for $k = 0$ to $\lceil \frac{b-a}{h} \rceil$ **do**
  $t_{k+1} = t_k + h$.
  $\hat{A}_{k+1} = h\beta A_{k+1} - \frac{1}{2} I$.
  $\hat{B}_{k+1} = \sqrt{h\beta} B_{k+1}$.
  $\hat{C}_{k+1} = [\sqrt{h\beta} C_{k+1}; \sqrt{-\alpha_0} Z_k^T; \ldots; \sqrt{-\alpha_{p-1}} Z_{k+1-p}^T]$.
  for $j = 1$ to j_{max} **do**
    Determine (sub)optimal ADI shift parameters $p_1^J, p_2^J, \ldots$ with respect to the matrix $F^j = \hat{A}_{k+1} - K^j \hat{B}_{k+1}^T$.
    $G^j = [\hat{C}_{k+1}^T \; K^{j-1}]$.
    Compute Z^j by Algorithm 3.5 such that the low rank factor product $Z^j Z^{jT}$ approximates the solution of $F^{jT} X^j + X^j F^j = -G^j G^{jT}$.
    $K^j = Z^j (Z^{jT} B)$.
  end for
  $Z_{k+1} = Z^{j_{max}}$.
end for

for $\psi_j(k) := t_k - t_{k-j}$. Having the local truncation error for the BDF methods expressed as (3.6) will allow us to compute it directly for low rank factors approximating the solution of DREs.
Note that if we want to vary the step and order of a linear multistep method 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 3.2.

Algorithm 3.2 Neville's Algorithm

Require: $\{(t_i, X_i)\}_{0\le i\le n}$, $t_i \in I \subset \mathbb{R}$, $X_i \approx X(t_i) \in \mathbb{R}^{n\times n}$.
$T_{i,o} := X_i \qquad 0 \le i \le n.$
$T_{i,k} := \frac{(t-t_{i-k})T_{i,k-1} - (t-t_i)T_{i-1,k-1}}{t_i - t_{i-k}} \qquad 0 \le i < k \le n.$

Assuming that

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

we get

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

so that

$$Z_{i,k} = \left[\sqrt{\frac{t-t_{i-k}}{t_i-t_{i-k}}}Z_{i,k-1} \qquad \sqrt{\frac{t-t_i}{t_{i-k}-t_i}}Z_{i-1,k-1}\right].$$

Hence Algorithm 3.2 can be written in terms of the low rank factors LRFs, see Algorithm 3.3.

Algorithm 3.3 LRF Neville's Algorithm

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

1: $Z_{i,o} := Z_i \qquad 0 \le i \le n.$

2: $Z_{i,k} := \left[\sqrt{\frac{t-t_{i-k}}{t_i-t_{i-k}}}Z_{i,k-1} \qquad \sqrt{\frac{t-t_i}{t_{i-k}-t_i}}Z_{i-1,k-1}\right], \qquad 0 \le i < k \le n.$

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

$$Z_{k,k} = [\sqrt{\lambda_0}Z_{0,0} \quad \sqrt{\lambda_1}Z_{1,0} \quad \dots \quad \sqrt{\lambda_k}Z_{k,0}].$$

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

$$Z_{2,2} = [\sqrt{\alpha_{220}\alpha_{110}}Z_{0,0} \quad \sqrt{-(\alpha_{020}\alpha_{221}+\alpha_{220}\alpha_{010})}Z_{1,0} \quad \sqrt{\alpha_{020}\alpha_{121}}Z_{2,0}]$$

where

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

Algorithm 3.3 will in general generate complex factors. However, we can still get real factor as solutions of the DRE in every step rewriting

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

where Z_p, Z_n are formed grouping the positive and negative $\lambda' s$ 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 3.4.

Algorithm 3.4 LRF Divided differences

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

1: $Z_{i,o} := Z_i \qquad 0 \le i \le 0.$

2: $Z_{i,k} := \left[\sqrt{\frac{1}{t_i-t_{i-k}}}Z_{i,k-1} \qquad \sqrt{\frac{1}{t_{i-k}-t_i}}Z_{i-1,k-1}\right] \quad 0 \le i < k \le 0.$

Analogous to Algorithm 3.3, Algorithm 3.4 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 cheaply 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.

Variable-coefficient BDF methods. 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. We will see this in the following.
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.

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

If the coefficients are computed in every step, changing the step size and the order of the method can be performed as for the single step methods. We review a strategy for this in the next section.
The variable coeffients for the second order BDF method are

$$\begin{aligned} \tilde{\beta}_0 &= \frac{h_k + h_{k-1}}{2h_k + h_{k-1}}, \\ \tilde{\alpha}_0 &= 1, \\ \tilde{\alpha}_1 &= -\left(\frac{h_k + h_{k-1}}{2h_k + h_{k-1}}\right)\left(1 + \frac{h_k}{h_k + h_{k-1}}\left(1 + \frac{h_k}{h_{k-1}}\right)\right), \\ \tilde{\alpha}_2 &= \left(\frac{h_k + h_{k-1}}{2h_k + h_{k-1}}\right)\left(\frac{h_k}{h_{k-1}}\right)\left(\frac{h_k}{h_k + h_{k-1}}\right). \end{aligned}$$

where h_k, h_{k-1}, are the step sizes. The variable coeffients for the third order BDF method can be found in [34].
An adaptive algorithm for the BDF methods can be implemented similar to the one which underlies the program DASSL of L.R. Petzold [38]. There the error estimators, which we used to decide whether to accept the current step or to redo this with a smaller step size can be computed by (3.6) instead of using the predictor polynomials involving the steps $p-1$, p, $p+1$; see [15, Algorithm on p. 373].

3.2. Lyapunov equation solver. For the problems we consider the spectrum of the solution often decays to zero rapidly. There are partial results explaining the decay of the eigenvalues of Lyapunov and Riccati solutions; bounds and estimates for the decay are given in [2, 36].

Methods based on this observation are proposed in [6, 35]. We will briefly describe them here. Let us consider Lyapunov equations of the form

$$F^T Y + Y F = -WW^T$$

with F stable, and $W \in \mathbb{R}^{n \times n_w}$. The *alternating directions implicit (ADI)* iteration applied to (3.2) yields

$$\begin{aligned} (F^T + p_j I) Y_{j-\frac{1}{2}} &= -WW^T - Y_{j-1}(F - p_j I), \\ (F^T + \overline{p_j} I) Y_j^T &= -WW^T - Y_{j-\frac{1}{2}}(F - \overline{p_j} I), \end{aligned} \tag{3.8}$$

where $\overline{p}$ denotes the complex conjugate of $p \in \mathbb{C}$, see [42] . If the shift parameters p_j are chosen appropriately, then $\lim_{j\to\infty} Y_j = Y$ with a superlinear convergence rate. An efficient way to compute these parameters is described in [10]. Starting this iteration with $Y_0 = 0$ and observing that for stable F, Y is positive semidefinite, it follows that $Y_j = Z_j Z_j^T$ for some $Z_j \in \mathbb{R}^{n \times r_j}$. Inserting this factorization into the above iteration, re-arranging terms and combining two iteration steps, we obtain a *factored* ADI iteration that in each iteration step yields n_w new columns of a full rank factor of Y (see [6, 33, 35] for several variants of this method). The method is described in Algorithm 3.5.

Algorithm 3.5 LRF ADI iteration

Require: F, W and set of ADI parameters $\{p_1, \ldots, p_k\}$
Ensure: $Z = Z_{i_{max}} \in \mathbb{C}^{n, i_{max} n_\omega}$ such that $ZZ^T \approx Y$.

1: $V_1 = \sqrt{-2\mathrm{Re}\,(p_1)}(F^T + p_1 I)^{-1} W$
2: $Z_1 = V_1$
3: **for** $j = 2, 3, \ldots$ **do**
4: $\quad V_j = \dfrac{\sqrt{\mathrm{Re}(p_j)}}{\sqrt{\mathrm{Re}(p_{j-1})}} \left(I - (p_j + \overline{p_{j-1}})(F^T + p_j I)^{-1}\right) V_{j-1}$
5: $\quad Z_j = \begin{bmatrix} Z_{j-1} & V_j \end{bmatrix}$
6: **end for**

It should be noted that all V_j's have the same number of columns as $W \in \mathbb{R}^{n \times n_w}$, i.e., at each iteration j, we have to solve w linear systems of equations with the same coefficient matrix $F^T + p_j I$. If convergence of the factored ADI iteration with respect to a suitable stopping criterion is achieved after $i_{\max}$ steps, then $Z_{i_{\max}} = [V_1, \ldots, V_{i_{\max}}] \in \mathbb{R}^{n \times i_{\max} n_w}$, where $V_j \in \mathbb{R}^{n \times n_w}$. For large n and small n_w we therefore expect that $r_i := i_{\max} n_w \ll n$. In that case, we have computed a low rank approximation $Z_{i_{\max}}$ to a factor Z of the solution, that is $Y = ZZ^T \approx Z_{i_{\max}} Z_{i_{\max}}^T$. In case, $n_w \cdot i_{\max}$ becomes large, a column compression technique [13, 23] can be applied to reduce the number of columns in $Z_{i_{\max}}$ without adding a significant error. Note that if the tolerance of the rank-revealing QR factorization is chosen according to the order of the method and the current step size we can apply a column compression technique without adding a significant error, [13]. This is not the case if QR factorization with normal pivoting strategy is applied. There the error that we are introducing can not be controlled.

Stopping criteria for the modified ADI iteration can be based either on the fact that $\|V_j\| \to 0$ very rapidly or on the residual norm $\|FZ_jZ_j^T + Z_jZ_j^T F^T + WW^T\|$; see [36] for an efficient way to compute the Frobenius norms of the residuals. On the

other hand, the Newton iteration inside is usually stopped when

$$\frac{\left\|Z_{j+1}Z_{j+1}^T - Z_jZ_j^T\right\|}{\left\|Z_jZ_j^T\right\|} < \tau$$

after few iterations, for a given tolerance threshold τ . However, this criterion is difficult to evaluate as it requires the explicit formation of iterates X_j. To overcome this difficulty we use a modified stopping criterion proposed in [4]. This criterion can be efficiently evaluated in case we use the Frobenious norm and the number of columns of the factors is much smaller than n. Moreover, the stopping criteria should be based on the tolerance for solving the differential equation.

3.3. Low rank Rosenbrock method. 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 (DIRK) 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 [15, 24]. 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. We focus on solving autonomous DREs by an efficient implementation of Rosenbrock methods based on a low rank version of the alternating direction implicit (ADI) described in the previous section. Although we just describe here the linearly implicit Euler method and a method of order two, the ideas in this can be straightforwardly applied to higher order Rosenbrock methods.

Linearly implicit Euler method. The one stage Rosenbrock method applied, as a matrix valued algorithm, to autonomous DREs of the form (3.1) can be written as

$$\begin{aligned} \bar{A}_k^T K_1 + K_1 \bar{A}_k &= -F(X_k) - hF_{t_k}, \\ X_{k+1} &= X_k + K_1, \end{aligned} \tag{3.9}$$

where $\bar{A}_k = A_k - R_kX_k - \frac{1}{2h}I$ and F defined as in (3.2). Moreover, (3.9) can be re-written such that the next iterate is computed directly from the Lyapunov equation, i.e.,

$$\bar{A}_k^T X_{k+1} + X_{k+1}\bar{A}_k = -Q - X_kSX_k - \frac{1}{h}X_k. \tag{3.10}$$

The right hand side of (3.10) is simpler to evaluate than the one in (3.9), so the implementation of (3.10) is more efficient [9]. If we assume,

$$\begin{aligned} Q = C^TC, &\qquad C \in \mathbb{R}^{p\times n}, \\ S = BB^T, &\qquad B \in \mathbb{R}^{n\times m}, \\ X_k = Z_kZ_k^T, &\qquad Z_k \in \mathbb{R}^{n\times z_k}. \end{aligned} \tag{3.11}$$

with p, m, $z_k \ll n$ and denoting $N_k = [\, C^T \;\; Z_k(Z_k^TB) \;\; \sqrt{h^{-1}}Z_k \,]$, then the Lyapunov equation (3.10) results in

$$\bar{A}_k^T X_{k+1} + X_{k+1}\bar{A}_k = -N_kN_k^T, \tag{3.12}$$

where $\bar{A}_k = A - B(Z_k(Z_k^T B))^T - \frac{1}{2h}I$. Observing that $\text{rank}(N_k) \leq p + m + z_k \ll n$, we can use the modified version of the *alternating directions implicit* (ADI) iteration to solve (3.12). The application of Algorithm 3.5 to (3.12) 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 $z_{k+1} \ll n$. This is described in Algorithm 3.6, for an equidistant mesh.

Algorithm 3.6 LRF linearly implicit Euler method

Require: $A \in \mathbb{R}^{n\times n}$, B, C, Z_0 satisfying (3.11), $t \in [a, b]$, 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: $\quad \bar{A}_k = A - B(Z_k(Z_k^T B))^T - \frac{1}{2h}I$.
4: $\quad N_k = [\, C^T \;\; Z_k(Z_k^T B) \;\; \sqrt{h^{-1}} Z_k \,]$.
5: $\quad$ Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$ with respect to the matrix $\bar{A}_k$.
6: $\quad$ Compute Z_{k+1} by Algorithm 3.5 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: $\quad t_{k+1} = t_k + h$.
8: **end for**

Rosenbrock method of second order. Let us now turn our attention to a second order method originally proposed in [14]. There the method is applied to atmospheric dispersion problems describing photochemistry, advective, and turbulent diffusive transport. As explained in [9] for solving autonomous DREs the method can be efficiently computed as:

$$X_{k+1} = X_k + \frac{3}{2}hK_1 + \frac{1}{2}hK_2, \tag{3.13}$$

$$\bar{A}_k^T K_1 + K_1 \bar{A}_k = -F(X_k), \tag{3.14}$$

$$\bar{A}_k^T K_{21} + K_{21}\bar{A}_k = -h^2 K_1 S K_1 + \left(\frac{1}{h\gamma} + 2\right)K_1, \tag{3.15}$$

$$K_2 = K_{21} + (1-h)K_1, \tag{3.16}$$

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 (3.13) and (3.16) can be computed directly as one step iteration like

$$X_{k+1} = X_k + \left(\frac{2h + h^2}{2}\right)K_1 + \frac{1}{2}hK_{21}, \tag{3.17}$$

As for the linearly implicit Euler method we want to apply the ADI iteration to solve the Lyapunov equations (3.14) and (3.15). Once again let us assume (3.11) and note that,

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

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,

$$F(X_k) = U_k U_k^T - N_k N_k^T, \tag{3.18}$$

thus, in order to solve (3.14) we can split the Lypunov equation as in Remark 3.1. The second equation (3.15) 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 3.7. There, Steps 6. and 8. of Algorithm 3.7 can be computed simultaneously by the factored ADI iteration as the linear systems of equations to be solved in each step have the same coefficient matrices.

Algorithm 3.7 LRF Rosenbrock method of second order

Require: $A \in \mathbb{R}^{n\times n}$, B, C, Z_0 satisfying (3.11), $t \in [a, b]$, 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 $\tilde{T}_1^k$ by Algorithm 3.5 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 $\hat{T}_1^k$ by Algorithm 3.5 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 by Algorithm 3.5 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**

REMARK 3.2. *The computation of Algorithm 3.7 is performed in complex arithmetics as in steps 9. and 10. complex factors are computed. 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* $X_k \approx Z_0 Z_0^T + h(\hat{T}_k \hat{T}_k^T - \tilde{T}_k \tilde{T}_k^T)$. *In order to do this, (3.15) is split in two Lyapunov equations,* $\bar{N}_k$ *is redefined and* $\bar{U}_k$ *has to be defined; these two matrices contain the positive and negative factors, respectively. However, the number of columns of them increases considerably in each step. Even though a column compression technique could be applied the convergence of the ADI iteration slows down considerably [34].*

Note that adaptive code for the method of order two described below, 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.

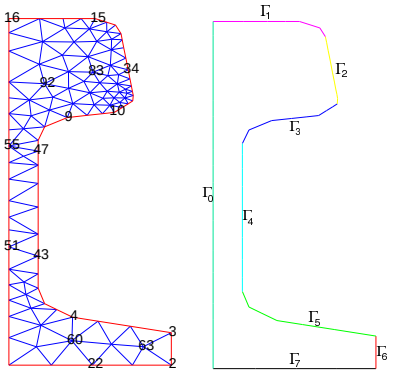

FIGURE 4.1. *Initial mesh with points of minimization (left) and partition of the boundary (right).*

The non-autonomous case. So far we have presented low rank versions of the Rosenbrock methods for autonomous DREs. We will now see that they can easily be extended to the non-autonomous case. In this case, $\frac{\partial F}{\partial t}$ appears, so 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} 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 DREs arising in optimal control the matrix $Q(t)$ is generally constant, it represents the output matrix), then F_{t_k} can be approximated by

$$\begin{aligned} F_{t_k} \approx \tfrac{1}{2h}\Big[& (Q_{k+1} - Q_{k-1}) + 2hA_k^T F(X_k) + (A_{k+1}^T - A_{k-1}^T)X_k \\ & +2hF(X_k)A_k + X_k(A_{k+1} - A_{k-1}) - 2hF(X_k)S_kX_k \\ & -X_k(S_{k+1} - S_{k-1})X_k - 2hX_kS_kF(X_k) \Big]. \end{aligned} \tag{3.19}$$

By (3.18) we know that $F(X_k)$ can be expressed as a combination of low rank factor matrix products, then re-arranging terms we can obtain a low rank matrix representation of (3.19). Finally, we point out that although we have focused on two particular Rosenbrock methods the ideas described in this section can be straightforwardly applied to a general s-stage Rosenbrock method or to an embedded method, [9].

4. Numerical results.

Optimal cooling of steel profiles. Let us consider the problem of optimal cooling of steel profiles, [7, 40, 41]. 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 Figure 4.1. A (linearized) version of the model has the form

$$\begin{array}{rcll} c\varrho x_t(\xi,t) & = & \lambda\Delta x(\xi,t) & \text{in } \Omega \times (0,T), \\ -\lambda\partial_\nu x(\xi,t) & = & g_i(t,x,u) & \text{on } \Gamma_i \text{ where } i = 0,\ldots,7, \\ x(\xi,0) & = & x_0(\xi) & \text{in } \Omega, \end{array} \tag{4.1}$$

Test	n	Q	R	G	T_f	h
1	371	I	I	0	20	0.01
2	1357	I	I	0	20	0.01
3	5177	I	I	0	20	0.01
4	20209	I	I	0	20	0.01

TABLE 4.1
Parameters for cooling of steel profiles problem.

n	DRE
371	9.1375 e+07
1357	5.0823 e+07
5177	4.0613 e+07
20209	3.9508 e+07

TABLE 4.2
Cost functional values for the fine time horizon (DRE).

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.

We applied the BDF method of order one with fixed step size for $n = 371, 1357$; and for the refined meshes $n = 5177, 20209$, the linearly implicit Euler method (Rosenbrock method of order one) was applied.
The problem parameters chosen can be found in Table 4.1. There n is the dimension, Q, R, G are the operators from the finite-dimensional LQR problem and h is the step size. We can see the behavior of the control parameter Γ_6 over time in Figure 4.2 for $n = 1357, 5177$. They converge to zero because $G = 0$ and therefore the final feedback matrix as well as the control are equal zero. In Table 4.2 the cost functional values are shown. Finally, in Figure 4.3 the convergence of the functional cost, which is defined in terms of the DREs, over the mesh size is plotted.

5. Conclusions. The numerical treatment of linear quadratic regulator problems for parabolic partial differential equations on a finite-time horizon requires solving large-scale DREs resulting from the semi-discretization. In order to give us an approximation framework for the computation of the finite-dimensional Riccati equations we have shown the convergence of the infnite-dimensional Riccati operators to the finite-dimensional ones for the (non)autonomous case. Typically the coefficient matrices of the resulting DRE 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 version 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.

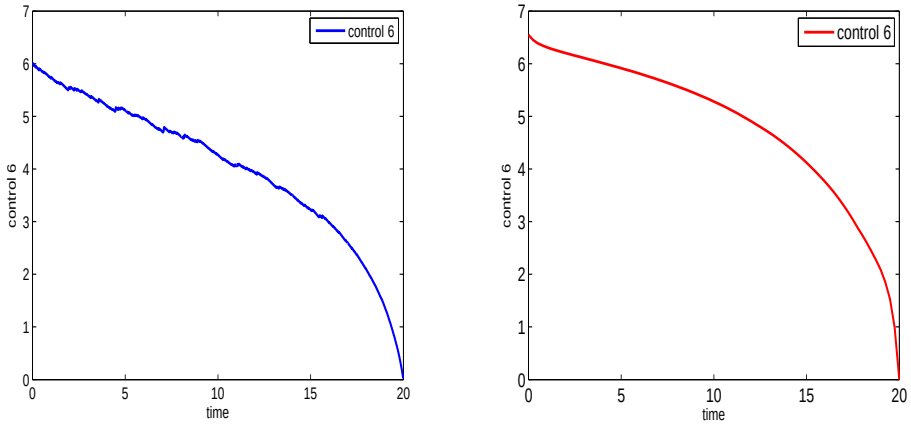

FIGURE 4.2. *Cooling of steel profiles control* Γ_6 *plotted over time for n=1357 4.2(a) and for n=5177 4.2(b).*

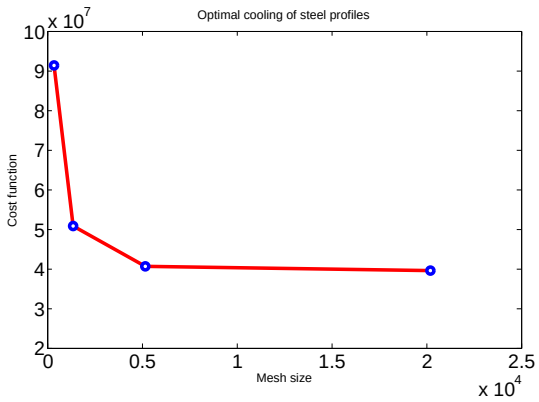

FIGURE 4.3. *Functional cost for different mesh sizes for the Optimal cooling of steel profiles example*

REFERENCES

[1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank. *Matrix Riccati Equations in Control and Systems Theory.* Birkhäuser, Basel, Switzerland, 2003. 1, 4, 9

[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(5):323–342, 2000. 10, 14

[3] U.M. Ascher and L.R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. PA. SIAM, Philadelphia, 1998. 10

[4] J. M. Badia, P. Benner, R. Mayo, and E. Quintana-Ortí. Parallel solution of large-scale and sparse generalized algebraic Riccati equations. In W. E. Nagel, W. V. Walter, and W. Lehner, editors, *Euro-Par 2006 Parallel Processing: 12th International Euro-Par Conference*, volume 4128 of *Lecture Notes in Computer Science*, pages 710–719. Springer-Verlag, 2006. 16

[5] H.T. Banks and K. Kunisch. The linear regulator problem for parabolic systems. *SIAM J. Cont. Optim.*, 22:684–698, 1984. 3, 5

[6] P. Benner, J.R. Li, and T. Penzl. Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems. *Numerical Linear Algebra with Applications*, 15(9):755–777, 2008. 11, 15

[7] P. Benner, V. Mehrmann, and D. Sorensen, editors. *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin/Heidelberg, Germany, 2005. 19

[8] P. Benner and H. Mena. BDF methods for large-scale differential Riccati equations. In B. De Moor, B. Motmans, J. Willems, P. Van Dooren, and V. Blondel, editors, *Proc. of Mathematical Theory of Network and Systems, MTNS 2004*, 2004. 10

[9] P. Benner and H. Mena. Rosenbrock methods for solving differential Riccati equations. Technical Report MPIMD/11-06, Max Planck Institute Magdeburg Preprints, October 2011. 16, 17, 19

[10] P. Benner, H. Mena, and J. Saak. On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations. *Electronic Transitions on Numerical Analysis*, 29:136–149, 2008. 15

[11] 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. 3

[12] 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. 3

[13] C.H. Bischof and G. Quintana-Ortí. Computing rank-revealing QR factorizations of dense matrices. *ACM Transactions on Mathematical Software*, 24(2):226–253, 1998. 15

[14] 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(4):1456–1480, 1999. 17

[15] F. Bornemann and P. Deuflhard. *Scientific Computing with Ordinary Differential Equations*, volume 42 of *Text in Applied Mathematics*. Springer-Verlag, New York, 2002. 14, 16

[16] C. Choi and A.J. Laub. Efficient matrix-valued algorithms for solving stiff Riccati differential equations. *IEEE Trans. Automat. Control*, 35:770–776, 1990. 1, 9

[17] R.F. Curtain and A. J. Pritchard. Infinite-dimensional Riccati equation for systems defined by evolution operators. *SIAM J. Cont. Optim.*, 14:951–983, 1976. 5, 7

[18] R.F. Curtain and T. Pritchard. *Infinite Dimensional Linear System Theory*, volume 8 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, New York, 1978. 2

[19] R.F. Curtain and H. Zwart. *An Introduction to Infinite-Dimensional Linear System Theory*. Texts in Applied Mathematics. Springer-Verlag, New York, 1995. 3

[20] L. Dieci. Numerical integration of the differential Riccati equation and some related issues. *SIAM J. Numer. Anal.*, 29(3):781–815, 1992. 1

[21] E. Eich. *Projizierende Mehrschrittverfahren zur numerischen Lösung von Bewegungsgleichungen technischer Mehrkörpersysteme mit Zwangsbedingungen und Unstetigkeiten*. PhD thesis, University of Augsburg, 1991. 11

[22] J.S. Gibson. The Riccati integral equation for optimal control problems in Hilbert spaces. *SIAM J. Cont. Optim.*, 17(4):537–565, 1979. 2, 3, 5, 6, 7, 8, 9

[23] S. Gugercin, D.C. Sorensen, and A.C. Antoulas. A modified low-rank Smith method for large-scale Lyapunov equations. *Numer. Algorithms*, 32(1):27–55, 2003. 15

[24] 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. 16

[25] S. Hein. *MPC/LQG-Based Optimal Control of Nonlinear Parabolic PDEs*. PhD thesis, TU Chemnitz, February 2010. available from http://nbn-resolving.de/urn:nbn:de:bsz:ch1-201000134. 2, 7

[26] K. Ito and K. Kunisch. Receding horizon optimal control for infinite dimensional systems. *ESAIM: Control Optim. Calc. Var.*, 8:741–760, 2002. 2

[27] K. Ito and K. Kunisch. Receding horizon control with incomplete observations. *SIAM J. Control Optim.*, 45(1):207–225, 2006. 2, 7

[28] O.L.R. Jacobs. *Introduction to Control Theory*. Oxford Science Publication, Oxford, 2nd edition, 1993. 1

[29] 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(5):1350–1385, 1991. 3

[30] I. Lasiecka and R. Triggiani. *Differential and Algebraic Riccati Equations with Application to Boundary/Point Control Problems: Continuous Theory and Approximation Theory*. Number 164 in Lecture Notes in Control and Information Sciences. Springer, Berlin, 1991. 2

[31] 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. 2, 3

[32] I. Lasiecka and R. Triggiani. *Control Theory for Partial Differential Equations: Continuous and Approximation Theories II. Abstract Hyperbolic-like Systems over a Finite Time Horizon*. Cambridge University Press, Cambridge, UK, 2000. 3

[33] J.R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002. 15

[34] H. Mena. *Numerical Solution of Differential Riccati Equations Arising in Optimal Control Problems for Parabolic Partial Differential Equations*. PhD thesis, Escuela Politecnica Nacional, 2007. 14, 18

[35] T. Penzl. *Numerische Lösung großer Lyapunov-Gleichungen*. PhD thesis, Technische Universität Chemnitz, 1998. 15

[36] T. Penzl. Eigenvalue decay bounds for solutions of Lyapunov equations: the symetric case. *Sys. Control Lett.*, 40:139–144, 2000. 10, 14, 15

[37] I.R. Petersen, V.A. Ugrinovskii, and A.V.Savkin. *Robust Control Design Using H^∞ Methods*. Springer-Verlag, London, UK, 2000. 1

[38] L. R. Petzold. A description of DASSL: A differential/algebraic system solver. *In Scientific Computing. North-Holland, Amsterdam, New York, London*, pages 65–68, 1982. 14

[39] 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.*, 25:121–144, 1987. 2

[40] J. Saak. Effiziente numerische Lösung eines Optimalsteuerungsproblems für die Abkühlung von Stahlprofilen. Diplomarbeit, Fachbereich 3/Mathematik und Informatik, Universität Bremen, D-28334 Bremen, September 2003. 19

[41] F. Tröltzsch and A. Unger. Fast solution of optimal control problems in the selective cooling of steel. *Z. Angew. Math. Mech.*, 81:447–456, 2001. 19

[42] E.L. Wachspress. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Letters*, 107:87–90, 1988. 15

[43] J. Zabczyk. Remarks on the algebraic Riccati equation. *Appl. Math. Optim.*, 2:251–258, 1976. 2