Numerical Solution of Differential Riccati Equations Arising in Optimal Control Problems for Parabolic Partial Differential Equations

by

Hermann Segundo Mena Pazmiño

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy at
Escuela Politécnica Nacional
partnership program with Technische Universität Berlin

July 2007

To my father

Abstract

The differential Riccati equations (DREs) arises in several applications, especially in control theory. Partial Differential Equations constraint optimization problems often lead to formulations as abstract Cauchy problems. Imposing a quadratic cost functional we obtain a linear quadratic regulator problem for an infinite-dimensional system. The optimal control is then given as the feedback control in terms of the operator differential Riccati equation. In order to solve such problems numerically we need to solve the large-scale DREs resulting from the semi-discretization. Typically the coefficient matrices of the resulting DRE have a given structure (e.g. sparse, symmetric, or low rank). We derive numerical methods capable of exploiting this structure. Moreover, we expect to treat stiff DREs, so we will focus on methods that can deal stiffness efficiently. Backward differentiation formulae (BDF) methods and Rosenbrock type methods are commonly used to solve stiff systems among linear multistep and one step ordinary differential equation (ODE) methods respectively. In this research we develop efficient matrix valued algorithms of these ODE methods suitable for large-scale DREs. The task of solving large-scale DREs appears also in nonlinear optimal control problems of tracking and stabilization type in the context of receding horizon techniques and model predictive control, i.e., we solve linearized problems on small time frames. We discuss the numerical solution of optimal control problems for instationary heat, convection-diffusion and diffusion-reaction equations formulating the problem as an abstract LQR problem.

Resumen

Las ecuaciones diferenciales de Riccati (EDRs) aparecen en muchas aplicaciones de ciencia e ingeniería, en especial en la teoría de control. Problemas de optimización gobernados por ecuaciones diferenciales parciales con frecuencia pueden formularse como problemas de Cauchy abstractos; si además se impone un funcional de costo cuadrático se obtiene un problema *linear quadratic regulator* para un sistema de dimensión infinita. La solución de este problema esta dada via *feedback* en términos de la ecuación diferencial de Riccati para operadores. De la semidiscretización de este problema resulta una ecuación matricial de Riccati a gran escala. Típicamente los coeficientes de la ecuación matricial resultante tienen una estructura definida (e.g., dispersión, simetría ó rango bajo). En este trabajo derivamos métodos numéricos capaces de explotar eficientemente esta estructura. Se espera que las EDRs sean rígidas (*stiff*), por lo que nos enfocaremos en métodos que puedan tratar el fenómeno de la rigidez eficientemente. Los métodos BDF (*backward differentiation formulae*) y los métodos de tipo Rosenbrock son comúnmente usados para tratar sistemas de ecuaciones diferenciales ordinarias (EDO) rígidos entre los métodos de multipaso y un paso, respectivamente. Por lo tanto derivamos versiones matriciales de estos algoritmos aplicables a EDRs a gran escala. El problema de resolver EDRs a gran escala es también de gran importancia en problemas de control óptimo no lineal de tipo *tracking* o *stabilization* en el contexto de *receding horizon* y *model predictive control*, i.e., se resuelven problemas lineales en intervalos de tiempo pequeños. En este marco estudiamos la resolución numérica de problemas de control óptimo para ecuaciones no estacionarias tales como: la ecuación del calor, convección-difusión formuladas previamente como problemas LQR abstractos.

Acknowledgements

Firstly, I would like to thank my supervisor, Professor Peter Benner, for providing me an interesting and challenging topic, for his many suggestions and constant support, and for giving me the opportunity to travel to Chemnitz University of Technology for research stays. The latter were crucial to the successful completion of this project.
I would also like to thank the staff of the Chemnitz working group *Mathematics in Industry and Technology*, Jens, René, Ulrike, Sabine, Heike for offering me their friendship and making my research stays there unforgettable experiences. I am particularly grateful to my dear friend Jens Saak together with whom (among the most interesting topics in life) big parts of this work were discussed in our mid-afternoon coffee breaks. Jens, thank you for the proof read of the manuscript and for helping to improve it by your clever comments.
My research would not have been possible without the invaluable support, patience, and love of my parents, Juanita and Marcco, my brother Ludwing, my sister Verónica, my nice Michelle and my beloved wife Cris. Particularly, I want to thank my father who has been my inexhaustible source of motivation and inspiration throughout my life.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Notation

$\mathbb{R}$:	set of real numbers
$\mathbb{C}$:	set of complex numbers
$\mathbb{R}^{n\times m}$:	space of $n \times m$ real matrices
$\mathbb{C}^{n\times m}$:	space of $n \times m$ complex matrices
$\mathbb{C}^{-}$:	the open left half plane of $\mathbb{C}$
I:	identity matrix
A^T:	transpose of matrix A
A^H:	hermitian of matrix A
rank(A):	rank of matrix A
A^{-1}:	inverse of A
$A > 0$:	positive definite
$A \geq 0$:	positive semidefinite
$\sigma(A)$:	spectrum of A
$\rho(A)$:	spectral radius of A
$\mathcal{PC}_m$[a,b]:	set of piecewise continuous functions $u(t) \in \mathbb{R}^m$, $t \in [a, b]$
$\mathcal{R}(z)$:	stability function of a numerical method for ordinary differential equations
$\mathbb{E}[.]$:	the expected value of a random variable
Φ_{JJ}:	the autocovariance of a stochastic process $J(t)$
cov(.) :	covariance matrix of a random variable
$\mathcal{L}(X, Y)$:	space of linear, bounded operators from a Banach space X to a Banach space Y, in case $Y = X$ we use $\mathcal{L}(X)$
$\mathcal{H}$:	state space
$\mathcal{U}$:	control space
$\mathcal{Y}$:	output space
$\|\cdot\|_{\mathcal{X}}$:	norm on space $\mathcal{X}$
$\langle\cdot,\cdot\rangle_{\mathcal{X}}$:	the duality product, or the inner product on $\mathcal{X}$
$L^p(a, b; \mathcal{U})$:	the Banach space of strongly measurable $\mathcal{U}$-valued functions $u(.)$ for which $\int_a^b \|u(t)\|^p\, dt < \infty^{\dagger}$, $L^2(a, b; \mathcal{U})$ is a Hilbert space with the inner product $\langle u_1(\cdot), u_2(\cdot)\rangle_{L^2} = \int_a^b \langle u_1(t), u_2(t)\rangle_{\mathcal{U}} dt$
$T(t)$:	one parameter semigroup, $t \geq 0$

$U(t,s)$:	strongly continuous evolution family, $t, s \in \mathbb{R}, t \geq s$
$\mathbf{A}$:	we use **bold** letters for infinite-dimensional operators and regular letters for the finite-dimensional ones
$\mathbf{A}^*$:	the Hilbert space adjoint of $\mathbf{A}$
$\mathrm{dom}(\mathbf{A})$:	domain of $\mathbf{A}$
∇:	nabla operator, $\nabla f = (\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n})$
Δ:	Laplace operator, $\Delta f = \nabla.(\nabla f) = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$

† for the definition of $L^p(a, b; \mathcal{U})$, as well as for the setting of optimal control problems in Hilbert spaces, the integral involved is the Bochner integral, see for instance [61].

CHAPTER

ONE

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.(see e.g. [2, 63, 69, 95]). In the literature there is a large variety of approaches to compute the solution of the DRE (see e.g. [38, 45, 46, 70]), however none of these methods seem to be suitable for large-scale control problems, since the computational effort grows at best like n^3, where n is the dimension of the state of the control system. In this thesis we consider the numerical solution of large-scale DREs arising in optimal control problems for parabolic partial differential equations. Hence, let us 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 $\mathbf{x}(\xi, t)$. For instance, in the problem of optimal cooling of steel profiles [24, 25, 49, 103, 112], $\mathbf{x}(\xi, t)$ denotes the temperature in ξ at time t, the convective term $\mathbf{c}$ as well as the uncontrolled reaction term $\mathbf{q}$ are equal to zero, and the diffusive part $\mathbf{k}$ depends on the material parameters: heat conductivity, heat capacity and density.

Moreover, we consider 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 $\xi \in \Omega$. Let $\hat{J}(\mathbf{x}, \mathbf{u})$ be a given performance index, then the control problem is given as:

$$\min_{\mathbf{u}} \hat{J}(\mathbf{x}, \mathbf{u}) \quad \text{subject to (1.1).} \tag{1.2}$$

If (1.1) is in fact linear, then a variational formulation leads to an abstract

Cauchy problem for a linear evolution equation of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \qquad \mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{H}, \tag{1.3}$$

for linear operators

$$\begin{aligned} \mathbf{A}: &\quad \mathrm{dom}(\mathbf{A}) \subset \mathcal{H} \to \mathcal{H}, \\ \mathbf{B}: &\quad \mathcal{U} \to \mathcal{H}, \\ \mathbf{C}: &\quad \mathcal{H} \to \mathcal{Y}, \end{aligned} \tag{1.4}$$

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 are only a finite number of independent control inputs to (1.1). Here $\mathbf{C}$ maps the states of the system to its outputs, such that

$$\mathbf{y} = \mathbf{C}\mathbf{x}. \tag{1.5}$$

If (1.1) is nonlinear, model predictive control technics can be applied [18, 67, 68]. There the equation is linearized at certain working points or around reference trajectories and linear problems for equations as in (1.3) have to be solved on subintervals of $[0, T_f]$. We review this technique in Chapter 6, Section 6.2.

In many applications in engineering the performance index $\hat{J}(\mathbf{x}, \mathbf{u})$ is given in quadratic form. We assume (1.3) to have a unique solution for each input $\mathbf{u}$ so that $\mathbf{x} = \mathbf{x}(\mathbf{u})$. Thus we can write the cost functional as $J(\mathbf{u}) := \hat{J}(\mathbf{x}(\mathbf{u}), \mathbf{u})$. Then

$$J(\mathbf{u}) = \frac{1}{2} \int_0^{T_f} \langle \mathbf{x}, \mathbf{Q}\mathbf{x} \rangle_{\mathcal{H}} + \langle \mathbf{u}, \mathbf{R}\mathbf{u} \rangle_{\mathcal{U}} \, dt + \langle \mathbf{x}_{T_f}, \mathbf{G}\mathbf{x}_{T_f} \rangle_{\mathcal{H}}, \tag{1.6}$$

where $\mathbf{Q}$, $\mathbf{G}$ are self-adjoint operators on the state space $\mathcal{H}$, $\mathbf{R}$ is a self-adjoint operator on the control space $\mathcal{U}$ and $\mathbf{x}_{T_f}$ denotes $\mathbf{x}(., T_f)$. To guarantee unique solvability of the control problem $\mathbf{R}$ is assumed positive definite. Since 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 $\mathbf{G}$. In many applications one simply has $\tilde{\mathbf{Q}} = \mathbf{I}$.

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 (see [40, 52, 76, 118]). We then have to consider the operator Riccati equations

$$0 = \Re(\mathbf{X}) := \mathbf{C}^* \mathbf{Q} \mathbf{C} + \mathbf{A}^* \mathbf{X} + \mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{X} \tag{1.7}$$

and

$$\dot{\mathbf{X}} = -\Re(\mathbf{X}) \tag{1.8}$$

depending on whether $T_f < \infty$ (1.8) or not (1.7). If $T_f = \infty$, then $\mathbf{G} = 0$ and the linear operator $\mathbf{X}$ is the solution of (1.7), i.e. $\mathbf{X} : \operatorname{dom}\mathbf{A} \to \operatorname{dom}\mathbf{A}^*$ and $\langle \hat{\mathbf{x}}, \Re(\mathbf{X})\mathbf{x}\rangle = 0$ for all $\mathbf{x}, \hat{\mathbf{x}} \in \operatorname{dom}(\mathbf{A})$. The optimal control is then given as the *feedback control*

$$\mathbf{u}_*(t) = -\mathbf{R}^{-1}\mathbf{B}^*\mathbf{X}_\infty\mathbf{x}_*(t), \tag{1.9}$$

which has the form of a regulator or closed-loop control. Here, $\mathbf{X}_\infty$ is the minimal nonnegative self-adjoint solution of (1.7), $\mathbf{x}_*(t) = S(t)\mathbf{x}_0(t)$, and $S(t)$ is the strongly continuous semigroup generated by $\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{X}_\infty$. In problems where $T_f < \infty$, the optimal control is defined similarly to (1.9), but then $\mathbf{X}_\infty$ represents the unique nonnegative solution of the differential Riccati equation (1.8) with terminal condition $\mathbf{X}_{T_f} = \mathbf{G}$ and therefore depends on time, i.e., it has to be replaced by $\mathbf{X}_\infty(t)$ in (1.9). Most of the required conditions, particularly the restrictive assumption that $\mathbf{B}$ is bounded, can be weakened [75, 76, 99]. In this thesis we will focus on the finite-time horizon case, $T_f < \infty$.

In order to solve the infinite-dimensional LQR problem numerically we use a Galerkin projection of the variational formulation of the PDE (1.1) onto a finite-dimensional space $\mathcal{H}^N$ spanned by a finite set of basis functions (e.g., finite element ansatz functions).

If we now choose the space of test functions as the space generated by finite element (fem) ansatz functions for a finite element semi-discretization in space, then the operators above have matrix representations in the fem basis. So we have to solve the discrete problem

$$\min_{u\in L^2(0,T_f;\mathcal{U})} \frac{1}{2}\int_0^{T_f} \langle x, Qx\rangle_{\mathcal{H}^N} + \langle u, \mathbf{R}u\rangle_{\mathcal{U}}\, dt + \langle x_{T_f}, Gx_{T_f}\rangle_{\mathcal{H}^N}, \tag{1.10}$$

with respect to

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ x(0) &= P^N\mathbf{x}_0, \\ y &= Cx. \end{aligned} \tag{1.11}$$

Here P^N is the projection operator from the space discretization method (here fem). Approximation results in terms of the Riccati solution operator $\mathbf{X}$ and the solution semigroup $S(t)$ for the closed loop system, validating this technique have been considered, e.g., in [12, 24, 65, 76, 87, 88]. Note that the control space is considered finite-dimensional and therefore does not change under spatial semi-discretization, i.e., we can directly apply the control computed for the discretized systems (1.11) to the infinite-dimensional system (1.3), although it might be suboptimal there. The estimation of the sub-optimality of that approach will be considered elsewhere.

To apply such a feedback control strategy to PDE control, in the finite-time horizon case, we need to solve the large-scale DREs resulting from the semi-discretization. Typically the coefficient matrices of the resulting DRE have a given structure (e.g. sparse, symmetric, or low rank). We derive numerical methods capable of exploiting this structure. Moreover, we expect to treat stiff DREs, so we will focus on methods that can deal with stiffness efficiently. Backward differentiation formula (BDF) methods and Rosenbrock methods are commonly used to solve stiff systems among linear multistep and one step ordinary differential equation (ODE) methods, respectively. In this research we develop efficient matrix valued algorithms of these ODE methods suitable for large-scale DREs.

Besides the vast variety of linear-quadratic problems that can be solved if an efficient DRE solver is available, the task of solving large-scale DREs appears also to become an increasingly important issue in nonlinear optimal control problems of tracking type and stabilization problems for classes of nonlinear instationary PDEs. 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, e.g. [18, 67, 68].

We discuss the numerical solution of optimal control problems governed by systems of the form (1.1), formulating the problem as an abstract LQR problem. Solving this problem, on a finite-time horizon, immediately leads to the problem of solving large-scale DREs, which we solve using our approach. Finally, we study the nonlinear case applying MPC, i.e. we solve linearized problems on small time frames using LQG design.

The outline of this thesis is now described, see Figure 1.1. In Chapter 2, we briefly summarize the basic concepts for finite and infinite-dimensional optimal control and the numerical solution of ordinary differential equations. Then, in Chapter 3, for the finite-time horizon case, we present an approximation framework for computation of Riccati operators than can be guaranteed to converge to the Riccati operator in feedback control. After that, we will review the existing methods to solve DREs and investigate whether they are suitable for large-scale problems arising in LQR and LQG design for semi-discretized parabolic partial differential equations. Based on this review in Chapter 4, we present efficient matrix valued algorithms of the BDF and Rosenbrock methods for ODEs. The crucial question of suitable stepsize and order selection strategies is also addressed. Solving the DRE using BDF methods requires the solution of an ARE in every step. The Newton-ADI iteration is an efficient numerical method for this task. It includes the solution of a Lyapunov equation by a low rank version of the alternating direction implicit (ADI) algorithm in each iteration step. The application of an s stage Rosenbrock method to the DRE requires the solution of one Lyapunov equation in each stage, as for the BDF methods, we solve the Lyapunov equation by the low rank version of the ADI algorithm. The convergence of the ADI algorithm strongly depends on the set of shift parameters. Therefore, a new method for determining sets of shift parameters for the ADI algorithm is proposed at the end of this chapter. In Chapter 5 numerical ex-

amples illustrating the efficiency of our algorithms are presented. Applications to linear control problems as well as nonlinear ones are presented in Chapter 6. Finally, in Chapter 7, conclusions regarding the results achieved in this thesis are drawn, as well as some opportunities for future research.

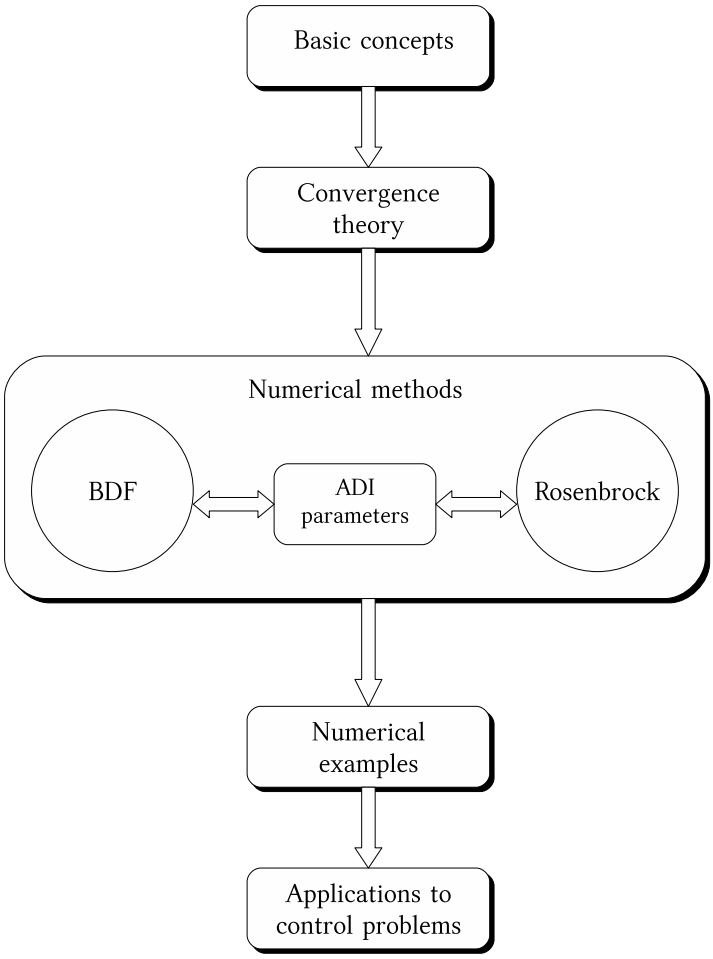


Figure 1.1: Guide to the thesis.

CHAPTER

TWO

Basic concepts

In this Chapter, we briefly summarize some concepts and results which we hope facilitates the reading of this thesis. First in section 2.1, basic concepts of the numerical solution of ordinary differential equations are presented. Then, in section 2.2 we review how the differential Riccati equation is involved in the solution of the finite-dimensional linear-quadratic optimal control problems. Existence and uniqueness results for the differential Riccati equation are presented also. Finally, a brief introduction to semigroup theory is given in section 2.3, as well as some results which are needed in Chapter 3.

2.1 Ordinary differential equations

Let us consider the following (ODE) ordinary differential equations system

$$\begin{array}{rcl} \dot{x} & = & f(t,x), \quad a \leq t \leq b \\ x(a) & = & x_a. \end{array} \tag{2.1}$$

The system (2.1) is said to be autonomous if f does not depend explicitly on time t, otherwise it is non-autonomous.
We discuss here stability and stiffness of ODEs, a detailed discussion can be found, e.g., in [7, 31, 57, 58].
The term stability has been used in a large variety of different concepts. It is important to be careful differentiating between stability of the system and stability of a numerical method for the system. We skip the stability of the system here, the interested reader can refer to specialized literature on the subject, see for instance [28].

Definition 2.1.1 *The function $\mathcal{R}(z)$, that can be interpreted as the numerical solution after one step for the famous* Dahlquist *test equation*

$$\dot{x} = \lambda x, \quad x_0 = 1, \quad z = h\lambda,$$

is called the stability function of the method. The set

$$S = \{z \in \mathbb{C} : |\mathcal{R}(z)| \leq 1\}$$

is called the stability domain of the method.

Example 2.1.2 :

(a) The stability function of the Euler method is:

$$\mathcal{R}(z) = 1 + z.$$

(b) The stability function of the Runge-Kutta method of order p is:

$$\mathcal{R}(z) = 1 + z + \frac{z^2}{2!} + \cdots + \frac{z^p}{p!} + \mathcal{O}(z^{p+1}).$$

Definition 2.1.3 *A method whose stability domain satisfies*

$$S \supset \mathbb{C}^- = \{z : \mathrm{Re}(z) \leq 0\}$$

is called A-stable.

A-stability is a desirable property of a numerical method to handle stiffness. However, it does not give a complete answer for this phenomenon. The trapezoidal rule and the midpoint rule as well (both have the same stability function) for the integration of first order ordinary differential equations is shown to posses (for a certain type of problem) an undesirable property, see Figure 2.1. To overcome this difficulty Ehle (1969) introduced the concept of L-stability.

Definition 2.1.4 *A method is called L-stable if it is A-stable and if in addition*

$$\lim_{z \to \infty} R(z) = 0.$$

We affirm that A-stability and specially L-stability are desirable properties to treat stiff problems. But, what exactly means that a system is stiff? We will briefly answer this question in the following.

2.1.1 Stiff systems

Stiffness does not have a universally accepted definition. Often it is described in terms of multiple time scaling. If the problem has widely varying time scales, and the phenomena that change on fast scales are stable, then the problem is stiff. In chemical reacting systems, stiffness often arises from the fact that some chemical reactions occur much more rapidly than others. In qualitative terms, (see [7] for a detail explanation) it could be defined as follows:

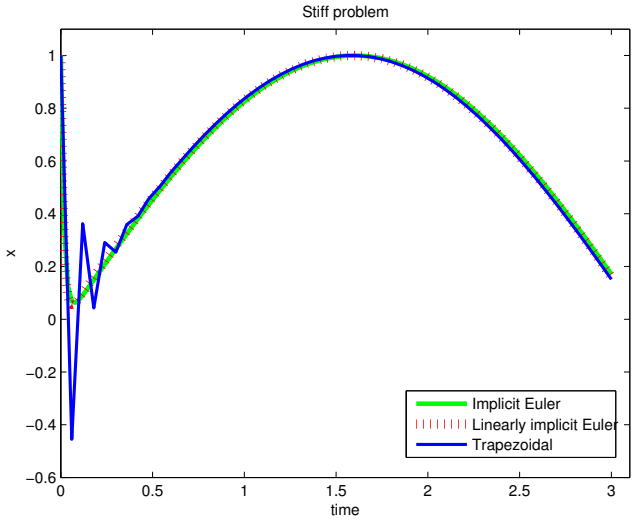


Figure 2.1: Approximated solution of (2.2).

Definition 2.1.5 *A system of the form (2.1) is stiff in some interval* $[a, b]$ *if the step size needed to maintain stability of the forward Euler method is much smaller than the step size required to represent the solution accurately.*

Example 2.1.6 Let us consider the stiff ODE system:

$$\begin{array}{rcl} \dot{x}(t) & = & -100(x(t) - sin(t)), \\ x(0) & = & 1 \quad t \geq 0. \end{array} \tag{2.2}$$

Figure 2.1 show the approximated solution of (2.2) by the implicit Euler method, the linearly implicit Euler method (Rosenbrock method of order one) and the implicit trapezoidal rule.

Notice that, in addition to the ODE system, stiffness depends on: the accuracy criterion, the length of the interval of integration, and the region of absolute stability of the method. Stiffness has to do with the ratio of eigenvalues and therefore, even though the concept of stiffness is best understood in qualitative terms, we could "define" stiffness as follows:

Remark 2.1.7 *A system of the form (2.1) is stiff if*

$$\frac{\max_i Re(\lambda_i)}{\min_i Re(\lambda_i)} \gg 1$$

where λ_i *are the eigenvalues of the Jacobian of* f *w.r.t.* x *and 100 could be taken as a "fuzzy boundary" between not being stiff and being stiff.*

Remark 2.1.7 can be useful in case the solution leaves a stiff domain and enters a non stiff domain making feasible the implementation of an integrator that switches from a method for stiff problems to one for non stiff problems, see [50].

2.2 Finite-dimensional LQR control theory

We will review the standard theory of the finite-dimensional optimal control theory for the finite-time horizon case. This theory can also be found in many textbooks, see for instance [5, 8, 35, 106]. We will closely follow the derivation in [14].
Let us consider the continuous time autonomous linear-quadratic optimal control problem

Minimize:

$$\mathcal{J}(u(.)) = \frac{1}{2}\int_0^{T_f} (y(t)^T Q y(t) + u(t)^T R u(t))dt \tag{2.3}$$

with respect to

$$\begin{array}{ll} \dot{x}(t) = Ax(t) + Bu(t), & t > 0, \quad x(0) = x_0, \\ y(t) = Cx(t), & t \geq 0, \end{array} \tag{2.4}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$ and $T_f < \infty$.

First of all, we will need some definitions and properties of the dynamical system (2.4).

Definition 2.2.1 *Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$.*

i) The matrix pair (A,B) is controllable if for all $x_1 \in \mathbb{R}^n$ there exists $t_1 \geq 0$ and $u \in \mathcal{PC}_m[0, t_1]$ such that $x(t_1) = x_1$.

ii) The matrix pair (C,A) is observable if the matrix pair (A^T, C^T) is controllable.

iii) The matrix pair (A,B) is stabilizable if for all x there exists u such that $\lim_{t\to\infty} x(t) = 0$ where x solves $\dot{x} = Ax + Bu$.

iv) The matrix pair (C,A) is detectable if x is the solution of $\dot{x} = Ax$ and $Cx(t) \equiv 0$ then $\lim_{t\to\infty} x(t) = 0$.

Proposition 2.2.2 *Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$.*

a) The following conditions are equivalent to the controllability of the matrix pair (A,B):

1. $\operatorname{rank}([B, AB, A^2B, \ldots, A^{n-1}B]) = n$ *(Hautus-Test).*

2. $\operatorname{rank}([A - \lambda I_n, B]) = n$ *for all* $\lambda \in \mathbb{C}$.

b) The following conditions are equivalent to the observability of the matrix pair (C,A):

1. $\text{rank}([C^T, (CA)^T, (CA^2)^T, \ldots, (CA^{n-1})^T]^T) = n$.
2. $\text{rank}([A^T - \lambda I, C^T]^T) = n$ *for all* $\lambda \in \mathbb{C}$.

c) The following conditions are equivalent to the stabilizability of the matrix pair (A,B):

1. $\text{rank}([A - \lambda I, B]) = n$ *for all* $\lambda \in \mathbb{C}$ *with* $Re(\lambda) \geq 0$.
2. *There exists* $K \in \mathbb{R}^{m \times n}$ *such that* $A + BK$ *is stable.*

d) The following conditions are equivalent to the detectability of the matrix pair (C,A):

1. *The matrix pair* (A^T, C^T) *is stabilizable.*
2. $\text{rank}([A^T - \lambda I, C^T]^T) = n$ *for all* $\lambda \in \mathbb{C}$ *with* $Re(\lambda) \geq 0$.
3. *There exists* $K \in \mathbb{R}^{n \times p}$ *such that* $A + KC$ *is stable.*

e) A matrix $K \in \mathbb{R}^{m \times n}$ *is stabilizing for* (A, B) *iff* $A + BK$ *is stable.*

Note that detectability and observability are dual concepts to controllability and stabilizability since the adjoint system of (2.4) is given by

$$\dot{x}(t) = A^T x(t) + C^T u(t), \tag{2.5}$$

with A and C as in (2.4).

2.2.1 Existence of solutions

Consider a cost functional given by

$$\mathcal{J}(u(.)) = \int_0^{T_f} g(t, x, u)dt$$

and a system described by the set of ordinary differential equations

$$\dot{x}(t) = f(t, x, u)$$

with initial condition $x(0) = x_0$ and no target condition for $x(T_f)$ is prescribed. In our case, the function g is given by

$$\begin{aligned} g(t, x, u) &\equiv g(x, u) \equiv g(x(t), u(t)) \\ &= \frac{1}{2}(x(t)^T C^T Q C x(t) + u(t)^T R u(t)) \\ &= \frac{1}{2}(y(t)^T Q y(t) + u(t)^T R u(t)), \end{aligned}$$

while the governing differential equation is defined via the function

$$f(t,x,u) \equiv f(x,u) \equiv f(x(t),u(t)) = Ax(t) + Bu(t).$$

Next, we define the Hamilton function by

$$\mathcal{H}(x,u,\mu) = -g(x,u) + \mu(t)^T f(x,u),$$

where the components of the co-state $\mu(t) \in \mathbb{R}^n$ satisfy $\dot{\mu}_j(t) = -\frac{\partial \mathcal{H}}{\partial x_j}$ for $j = 1,\ldots,n$, which is in our case equivalent to

$$\dot{\mu}(t) = C^T QCx(t) - A^T \mu(t). \tag{2.6}$$

From the Potryagin Maximum Principle for autonomous systems as given, e.g., in [98, Theorem 4.3], we obtain:

Proposition 2.2.3 *Let $u_*(t) \in \mathcal{PC}_m[0,T_f]$ and let x_* be the trajectory determined by $\dot{x}(t) = Ax(t) + Bu_*(t)$, $x(0) = x_0$. Then in order for u_* to be optimal, i.e, $\mathcal{J}(u_*) \leq \mathcal{J}(u)$ for all $u \in \mathcal{PC}[0,T_f]$, it is necessary that the following two conditions hold.*

(i) $\mathcal{H}(x,u_*,\mu) \geq \mathcal{H}(x,u,\mu)$ *on* $[0,T_f]$ *for all* $u \in \mathcal{PC}_m[o,T_f]$;

(ii) $\mu(T_f) = 0$.

Condition (i) is called the maximum condition while (ii) is a transversality condition.
As u is not constraint, we obtain from Proposition 2.2.3 (i) that $\frac{\partial \mathcal{H}}{\partial u_j} = 0$ for $j = 1,\ldots,m$ and hence it follows that

$$-Ru(t) + B^T \mu(t) = 0 \tag{2.7}$$

must hold on $[0,T_f]$ for an optimal control. Moreover, the second derivative test implies $R \geq 0$ as a necessary condition for the existence of an optimal control minimizing the objective functional $\mathcal{J}(u)$.
Collecting all equations, i.e, the state equations together with the initial conditions, (2.6) together with the transversality condition, and (2.7), we obtain

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), & x(0) = x_0, \\ \dot{\mu}(t) &= C^T QCx(t) - A^T \mu(t), & \mu(T_f) = 0, \\ 0 &= Ru(t) - B^T \mu(t). \end{aligned}$$

These equations can be combined to the two-point boundary value problem

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & I_n & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\mu} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} A & 0 & B \\ C^T QC & -A^T & 0 \\ 0 & -B^T & R \end{bmatrix} \begin{bmatrix} x \\ \mu \\ u \end{bmatrix}, \tag{2.8}$$
$$x(0) = x_0, \qquad \mu(T_f) = 0.$$

Note that $\dot{u}$ only appears formally, so that (2.8) does not require additional smoothness properties for u. Actually, (2.8) is a boundary value problem for a differential algebraic equation where the co-state μ and the control are related by a purely algebraic equation. Assuming R nonsingular, u can be removed from the system, yielding an ordinary boundary value problem; see next section.
Due to the special structure of the autonomous linear-quadratic optimal control problem, the conditions derived from the Pontryagin Maximum Principle yield necessary and sufficient conditions for existence of an optimal control. These are summarized in the following theorem, see e.g [35].

Theorem 2.2.4 *a) If $u_* \in \mathcal{PC}_m[0,T_f]$ is an optimal control for the linear-quadratic optimization problem (2.3)-(2.4), then there exists a co-state μ with $\mu(t) \in \mathbb{R}^n$ such that $[(x_*(t))^T,(u_*(t))^T,(\mu(t))^T]^T$ satisfies the two-point boundary value problem (2.8).*
b) If $[(x_*(t))^T,(u_*(t))^T,(\mu(t))^T]^T$ satisfies the two-point boundary value problem (2.8) and Q, R are positive semidefinite, then $\mathcal{J}(u_*) \leq \mathcal{J}(u)$ for all $u \in \mathcal{PC}_m[0,T_f]$ and for all (x,u) satisfying (2.4).

The above theorem yields conditions for the existence of a solution of the optimal control problem by transforming the constrained optimization problem to a boundary value problem.

2.2.2 Differential Riccati equations

Assuming that R is nonsingular (i.e, together with $R \geq 0$ this implies that R is positive definite, denoted here by $R > 0$), (2.7) is equivalent to

$$u(t) = R^{-1}B^T\mu(t), \tag{2.9}$$

such that the state equations can be written as

$$\dot{x}(t) = Ax(t) + Bu(t) = Ax(t) + BR^{-1}B^T\mu(t). \tag{2.10}$$

Using (2.10) the two point boundary value problem (2.8) can be written as

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\mu}(t) \end{bmatrix} = \begin{bmatrix} A & BR^{-1}B^T \\ C^TQC & -A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \mu(t) \end{bmatrix}, \qquad \begin{matrix} x(0) = x_0, \\ \mu(T_f) = 0. \end{matrix} \tag{2.11}$$

Making the *ansatz* $\mu(t) := -X(t)x(t)$, the terminal condition for the co-state transforms to $\mu(T_f) = X(T_f)x(T_f)$ which together with $\mu(T_f) = 0$, and the fact that $x(T_f)$ is unspecified implies $X(T_f) = 0$. Employing $\dot{\mu}(t) = -\dot{X}(t)x(t) - X(t)\dot{x}(t)$ we obtain from the first differential equation in (2.11)

$$\dot{x}(t) = Ax(t) - BR^{-1}B^TX(t)x(t),$$

while the second yields

$$\begin{aligned} C^TQCx(t) + A^TX(t)x(t) &= -\dot{X}(t)x(t) - X(t)\dot{x}(t) \\ &= -\dot{X}(t)x(t) - X(t)(Ax(t) - BR^{-1}B^TX(t)x(t)). \end{aligned}$$

The latter is equivalent to

$$(\dot{X}(t) + X(t)A + A^T X(t) - X(t)BR^{-1}B^T X(t) + C^T QC)x(t) = 0$$

for all $t \in]0, T_f[$. Hence, as $x(t)$ is unspecified, we obtain the matrix differential Riccati equation (DRE)

$$\dot{X}(t) = -(C^T QC + X(t)A + A^T X(t) - X(t)BR^{-1}B^T X(t)), \tag{2.12}$$

i.e., an autonomous nonlinear matrix-valued differential equation. Together with $X(T_f) = 0$ this yields an initial value problem in reverse time.
The existence and uniqueness of the DRE (2.12) is a direct consequence of [2, Thm. 4.1.6], which we cite below.

Theorem 2.2.5 *If $S(t)$, $Q(t) \geq 0$ for $t \leq t_0$, then the unique solution X of the Riccati differential equation*

$$\begin{aligned} \dot{X}(t) &= -Q(t) - A^*(t)X(t) - X(t)A(t) + X(t)S(t)X(t), \\ X(t_0) &= X_0 \geq 0, \end{aligned}$$

where $Q(t)$, $A(t)$, $R(t) \in \mathbb{C}^{n\times n}$ are piecewise continuous, locally bounded functions, exists for $t \leq t_0$ with

$$0 \leq X(t) \leq \tilde{X}(t) \quad for \quad t \leq t_0;$$

here $\tilde{X}$ is the solution of

$$\dot{\tilde{X}} = -A^*(t)\tilde{X}(t) - \tilde{X}(t)A(t) - Q(t), \quad \tilde{X}(t_0) = X_0.$$

A detailed discussion of the theory of Riccati equations can be found in many books, e.g., [2, 71, 101].
Transposing equation (2.12) we see that $X(t)^T$ has to satisfy the same differential equation as $X(t)$ on the whole interval $[0, T_f]$. From Theorem 2.2.5 it follows that $X_*(t) = X_*(t)^T$, i.e., the solution $X_*(t)$ is symmetric.
Under the given assumptions we obtain that the two-point boundary value problem (2.11) has a unique solution given by

$$\mu_*(t) = X_*(t)x_*(t), \qquad t \in [0, T_f],$$

where $x_*(t)$ is the unique solution of the linear initial value problem

$$\dot{x}(t) = (A - BR^{-1}B^T X_*(t))x(t), \qquad x(0) = x_0.$$

Summarizing all results, we obtain the following theorem.

Theorem 2.2.6 *If $Q \geq 0$, $R > 0$, and $T_f < \infty$, then there exists a unique solution of the linear-quadratic optimal control problem (2.3)-(2.4). The optimal control is given by the feedback law*

$$u_*(t) = -R^{-1}B^T X_*(t)x(t), \tag{2.13}$$

where $X_*(t)$ satisfies the DRE

$$\dot{X}(t) = -(C^T QC + X(t)A + A^T X(t) - X(t)BR^{-1}B^T X(t)),$$

with the terminal condition $X(T_f) = 0$. Moreover, for any initial value X_0 the optimal cost is

$$\mathcal{J}(u_*(.)) = \frac{1}{2}(x_0)^T X_*(0)x_0.$$

The optimal control is therefore given as a closed-loop control, i.e., the system state is used to determine the input via the feedback law (2.13). The matrix $K_*(t) := R^{-1}B^T X_*(t)$ is called the optimal gain matrix.

Remark 2.2.7 *Let X(t) be the solution of the DRE (2.12). Define $\tilde{X}(t, T_f) = X(T_f - t)$. Then $\tilde{X}$ satisfies the DRE*

$$\dot{\tilde{X}}(t) = C^T QC + \tilde{X}(t)A + A^T \tilde{X}(t) - \tilde{X}(t)BR^{-1}B^T \tilde{X}(t),$$

with the initial condition $\tilde{X}(0, T_f) = X(T_f) = 0$. Observing that

$$\lim_{T_f \to 0} \dot{\tilde{X}}(t, T_f) = 0$$

and denoting $X_\infty(t) := \lim_{T_f \to 0} \tilde{X}(t, T_f)$, then $X_\infty(t)$ satisfies the algebraic Riccati equation

$$0 = C^T QC + \tilde{X}_\infty(t)A + A^T \tilde{X}_\infty(t) - \tilde{X}_\infty(t)BR^{-1}B^T \tilde{X}_\infty(t).$$

As $X_\infty(t)$ has to satisfy the same equation for any $t \in [0, \infty[$, the solution is time-invariant, i.e., $X_\infty(t) \equiv X_\infty$.

2.3 Semigroup theory

In the following we will briefly summarize some basic concepts of semigroup theory as well as some results of the theory applied to the linear-quadratic control problem for infinite-dimensional systems. The theorems cited here will be particularly important to prove the convergence result proposed in Chapter 3.

2.3.1 Introduction

The theory of (one-parameter) semigroups of linear operators in Banach spaces started in the 1950s with the Hille-Yosida generation theorem. The theory is now a well known subject thanks to the efforts of many people. Particularly, semigroups have become an important tool for integro-differential equations and functional equations, in infinite-dimensional control theory, e.g. [44, 48, 53, 89]. Here we follow the book of Engel and Nagel [48], we keep their notation and skip the proofs of the theorems.

The idea behind semigroups is strongly related with the solution of an autonomous initial value problem. In 1821 Cauchy asks in his *Course d'Analyse*:

Determine the function $\varphi(x)$ in such a way that it remains continuous between two arbitrary real limits of the variable x, and that, for all real values of the variables x and y, one has

$$\varphi(x+y) = \varphi(x)\varphi(y).$$

The exponential functions solves the problem. In fact they are the only solutions of Cauchy's problem. The problem can be reformulated as:

Cauchy's problem. Find all maps $T(.) : \mathbb{R}_+ \to \mathbb{C}$ satisfying the functional equation

$$\begin{array}{l} T(t+s) = T(t)T(s) \quad \text{for all} \quad s,t \geq 0, \\ T(0) = 1. \end{array} \tag{2.14}$$

The property listed below will show how Cauchy's problem is related to an autonomous initial value problem.

Proposition 2.3.1 *Let $T(t) := e^{ta}$ for some $a \in \mathbb{C}$ and all $t \geq 0$. Then the function $T(.)$ is differentiable and satisfies the differential equation (or, more precisely, the initial value problem)*

$$\begin{array}{l} \frac{dT}{dt}(t) = aT(t) \quad \text{for all} \quad t \geq 0, \\ T(0) = 1. \end{array} \tag{2.15}$$

Conversely, the function $T(.) : \mathbb{R}_+ \to \mathbb{C}$ defined by $T(t) = e^{ta}$ for some $a \in \mathbb{C}$ is the only differentiable function satisfying (2.15). Finally, we observe that $a = \frac{dT}{dt}(t)|_{t=0}$.

Hence, the answer to Cauchy's problem is given by:

Theorem 2.3.2 *Let $T(.) : \mathbb{R}_+ \to \mathbb{C}$ be a continuous function satisfying (2.14). Then there exists a unique $a \in \mathbb{C}$ such that*

$$T(t) = e^{ta} \quad \text{for all } t \geq 0. \tag{2.16}$$

In the following we will see how the extention of this scalar problem to Banach spaces leads us to the definition of a semigroup.

2.3.2 Definitions and properties

First of all, we define a Banach algebra.

Definition 2.3.3 *A Banach algebra is an associative algebra E (i.e. a vector space which also allows the multiplication of vectors in a distributive and associative manner) over the real or complex numbers which at the same time is*

also a Banach space. The algebra multiplication and the Banach space norm are required to be related by the following inequality:

$$\|x\,y\| \ \leq \|x\|\,\|y\| \qquad \text{for all } x, y \in E$$

This ensures that the multiplication operation is continuous.

If we take X to be a complex Banach space with norm $\|.\|$ and denote by $\mathcal{L}(X)$ the Banach algebra of all bounded linear operators on X endowed with the operator norm. We can state Cauchy's problem in this context as:

Cauchy's problem on Banach spaces. Find all maps $T(.) : \mathbb{R}_+ \to \mathcal{L}(X)$ satisfying the functional equation

$$\begin{array}{l} T(t+s) = T(t)T(s) \quad \text{for all} \quad s, t \geq 0, \\ T(0) = I, \end{array} \tag{2.17}$$

where I represents the identity operator.

Definition 2.3.4 *A family $(T(t))_{t\geq 0}$ of bounded linear operators on a Banach space X is called a (one-parameter) semigroup (or linear dynamical system) on X if it satisfies the functional equation (2.17). If (2.17) holds even for all $t, s \in \mathbb{R}$, we call $(T(t))_{t\in\mathbb{R}}$ a (one-parameter) group on X.*

Let $\mathbf{A} \in \mathcal{L}(X)$, we define an operator-valued exponential function by

$$e^{t\mathbf{A}} := \sum_{k=0}^{\infty} \frac{t^k \mathbf{A}^k}{k!}, \tag{2.18}$$

where the convergence of the series takes place in the Banach algebra $\mathcal{L}(X)$. Then, similar to Proposition 2.3.1 the next result can be stated.

Proposition 2.3.5 *For $\mathbf{A} \in \mathcal{L}(X)$ define $(e^{t\mathbf{A}})_{t\geq 0}$ by (2.18). Then, the following properties hold.*

(i) $(e^{t\mathbf{A}})_{t\geq 0}$ is a semigroup on X such that the map

$$\mathbb{R}_+ \ni t \mapsto e^{t\mathbf{A}} \in (\mathcal{L}(X), \|.\|)$$

is continuous.

(ii) The map $\mathbb{R}_+ \ni t \to T(t) := e^{t\mathbf{A}} \in (\mathcal{L}(X), \|.\|)$ is differentiable and satisfies the differential equation

$$\begin{array}{l} \frac{dT}{dt}(t) = \mathbf{A}T(t) \quad \textit{for all} \quad t \geq 0, \\ T(0) = I. \end{array} \tag{2.19}$$

Conversely, every differential function $T(.) : \mathbb{R}_+ \to (\mathcal{L}(X), \|.\|)$ satisfying (2.19) is already of the form $T(t) = e^{t\mathbf{A}}$ for some $\mathbf{A} \in \mathcal{L}(X)$.

Finally we observe that $\mathbf{A} = \dot{T}(0)$.

Before giving a satisfactory answer to Cauchy's problem in Banach spaces, the concept of uniformly continuous semigroups is introduced.

Definition 2.3.6 *A one-parameter semigroup* $(T(t))_{t\geq 0}$ *on a Banach space* X *is called uniformly continuous (or norm continuous) if*

$$\mathbb{R}_+ \ni t \mapsto T(t) \in \mathcal{L}(X)$$

is continuous with respect to the uniform operator topology on $\mathcal{L}(X)$.

With this terminology an answer to Cauchy's problem can be stated as the following theorem.

Theorem 2.3.7 *Every uniformly continuous semigroup* $(T(t))_{t\geq 0}$ *on a Banach space* X *is of the form*

$$T(t) = e^{t\mathbf{A}}, \quad t \geq 0,$$

for some bounded operator $\mathbf{A} \in \mathcal{L}(X)$.

However, uniform continuity is in general too strong as a requirement for many semigroups defined on concrete function spaces. For instance, for a function $f : \mathbb{R} \to \mathbb{C}$ and $t \geq 0$, the operators $T_l(t)$ such that

$$(T_l(t)f)(s) := f(s + t), \quad s \in \mathbb{R}$$

are called the left translation (of f by t), while

$$(T_r(t)f)(s) := f(s - t), \quad s \in \mathbb{R}$$

are called the right translation (of f by t). The operators $T_l(t)$ define a one-parameter (semi)group the so called *translation* (semi)groups which are not uniformly continuous.
Instead *strong* continuity holds in most applications. Let us define a class of semigroups satisfying strong continuity.

Definition 2.3.8 *A family* $(T(t))_{t\geq 0}$ *of bounded linear operators on a Banach space* X *is called strongly continuous (one-parameter) semigroup (or* $\mathcal{C}_0$*-semigroup*[1]*) if (2.17) holds and the maps*

$$\xi_x : t \mapsto \xi_x(t) := T(t)x \tag{2.20}$$

are continuous from $\mathbb{R}_+$ *into* X *for every* $x \in X$.

The following result can be very useful to prove a semigroup to be strongly continuous.

Proposition 2.3.9 *For a semigroup* $(T(t))_{t\geq 0}$ *on a Banach space* X*, the following assertions are equivalent.*

[1] $\mathcal{C}_0$ abbreviates "Cesàro" summable of order 0,

(a) $(T(t))_{t\geq 0}$ *is strongly continuous.*

(b) $\lim_{t\downarrow 0} T(t)x = x$ *for all* $x \in X$.

(c) *There exists* $\delta > 0$, $M \geq 1$, *and a dense subset* $D \subset X$ *such that*

(i) $\|T(t)\| \leq M$ *for all* $t \in [0, \delta]$,

(ii) $\lim_{t\downarrow 0} T(t)x = x$ *for all* $x \in D$.

Proposition 2.3.10 *For a strongly continuous semigroup* $(T(t))_{t\geq 0}$, *there exist constants* $\omega \in \mathbb{R}$ *and* $M \geq 1$ *such that*

$$\|T(t)\| \leq Me^{\omega t}$$

for all $t \geq 0$.

Definition 2.3.11 *The (infinitesimal) generator* $\mathbf{A} : D(\mathbf{A}) \subset X \to X$ *of a strongly continuous semigroup* $(T(t))_{t\geq 0}$ *on a Banach space* X *is the operator*

$$\mathbf{A}x := \dot{\xi}_x(0) = \lim_{h\downarrow 0} \frac{1}{h}(T(h)x - x)$$

defined for every x *in its domain*

$$D(\mathbf{A}) := \{x \in X : \xi_x \text{ is differentiable}\}.$$

In order to retrieve the semigroup $(T(t))_{t\geq 0}$ from its generator $(\mathbf{A}, D(\mathbf{A}))$, a third object is needed the *resolvent*.

Definition 2.3.12 *Let* $(T(t))_{t\geq 0}$ *be a semigroup and* $\mathbf{A}$ *its generator* ($D(\mathbf{A}) \subset X$), *the resolvent operator*

$$R(\lambda, \mathbf{A}) := (\lambda - A)^{-1} \in \mathcal{L}(X)$$

is defined for all complex numbers in the resolvent $\rho(A)$, *where*

$$\rho(\mathbf{A}) := \{\lambda \in \mathbb{C} : \lambda - \mathbf{A} : D(\mathbf{A}) \to X \text{ is bijective}\}$$

(its complement $\sigma(\mathbf{A}) := \mathbb{C}\backslash\rho(\mathbf{A})$ *is the spectrum of* $\mathbf{A}$*).*

Theorem 2.3.13 *The generator of a strongly continuous semigroup is a closed and densely defined linear operator that determines the semigroup uniquely.*

A satisfactory answer to Cauchy's problem in terms of strongly continuous semigroups require much more effort than in case of uniformly continuous semigroups. For example, the characterization of linear operators that are the generators of strongly continuous semigroups requires the Hille-Yosida generation theorems. The interested reader is refered to [48] and references therein for a detailed explanation.
We finish this review defining a special class of semigroups, the analytic semigroups.

Definition 2.3.14 *A closed linear operator $(\mathbf{A}, D(\mathbf{A}))$ with dense domain $D(\mathbf{A})$ in a Banach space X is called sectorial (of angle δ) if there exists $0 \leq \delta \leq \frac{\pi}{2}$ such that the sector*

$$\Sigma_{\frac{\pi}{2}+\delta} := \left\{\lambda \in \mathbb{C} : |\arg \lambda| < \frac{\pi}{2} + \delta\right\} \backslash \{0\}$$

is contained in the resolvent set $\rho(A)$, and if for each $\varepsilon \in (0, \delta)$ there exists $M_\varepsilon \geq 1$ such that

$$\|R(\lambda, \mathbf{A})\| \leq \frac{M_\varepsilon}{|\lambda|} \quad \text{for all } \ 0 \neq \lambda \in \bar{\Sigma}_{\frac{\pi}{2}+\delta-\varepsilon}$$

Proposition 2.3.15 *Let $(\mathbf{A}, D(\mathbf{A}))$ be a sectorial operator of angle δ. Then for all $z \in \Sigma_{\frac{\pi}{2}+\delta}$, the maps $T(z)$ are bounded linear operators on X satisfying the following properties.*

(i) $\|T(z)\|$ *is uniformly bounded for* $z \in \Sigma_{\frac{\pi}{2}+\delta'}$ *if* $0 < \delta' < \delta$.

(ii) The map $z \mapsto T(z)$ *is analytic in* $\Sigma_{\frac{\pi}{2}+\delta}$.

(iii) $T(z_1 + z_2) = T(z_1)T(z_2)$ *for all* $z_1, z_2 \in \Sigma_{\frac{\pi}{2}+\delta}$.

(iv) The map $z \mapsto T(z)$ *is strongly continuous in* $\Sigma_{\frac{\pi}{2}+\delta'} \cup \{0\}$ *if* $0 < \delta' < \delta$.

Definition 2.3.16 *A family of operators $(T(z))_{z\in\Sigma_\delta\cup\{0\}} \subset \mathcal{L}(X)$ is called an analytic semigroup (of angle $\delta \in (0, \frac{\pi}{2}]$) if*

(i) $T(0) = I$ *and* $T(z_1 + z_2) = T(z_1)T(z_2)$ *for all* $z_1, z_2 \in \Sigma_\delta$.

(ii) The map $z \mapsto T(z)$ *is analytic in* Σ_δ.

(iii) $\lim_{\Sigma_{\delta'} \ni z \to 0} T(z)x = x$ *for all* $x \in X$ *and* $0 < \delta' < \delta$.

If, in addition,

(iv) $\|T(z)\|$ *is bounded in* $\Sigma_{\delta'}$ *for every* $0 < \delta' < \delta$,

we call $(T(z))_{z\in\Sigma_\delta\cup\{0\}}$ a bounded analytic semigroup.

Semigroups for Non-autonomous Cauchy Problems. For partial differential equations in which the coefficients are time-variant, the operators are time dependent. Therefore, we replace the fixed operator $\mathbf{A}$ by operators $\mathbf{A}(t)$ depending on a (time) parameter $t \in \mathbb{R}$. Similar to the time-invariant case the differential equation that has to be satisfied can be stated as

$$\begin{array}{l} \frac{du}{dt}(t) = \mathbf{A}(t)u(t) \quad \text{for all} \quad t, s \in \mathbb{R}, t \geq s, \\ u(s) = x. \end{array} \tag{2.21}$$

on a Banach space X. The problem becomes much more complicated. Let us first interpret what a solution of (2.21) means.

Definition 2.3.17 *Let $(\mathbf{A}(t), D(\mathbf{A}(t)))$, $t \in \mathbb{R}$, be linear operators on the Banach space X and take $s \in \mathbb{R}$ and $x \in D(\mathbf{A}(s))$. Then a (classical) solution of (2.21) is a function $u(.; s, x) = u \in C^1([s, \infty), X)$ such that $u(t) \in D(\mathbf{A}(t))$ and u satisfies (2.21) for $t \geq s$.*
The Cauchy problem (2.21) is called well-posed *(on spaces Y_t) if there are dense subspaces $Y_s \subset D(\mathbf{A}(s))$, $s \in \mathbb{R}$, of X such that for $s \in \mathbb{R}$ and $x \in Y_s$ there is a unique solution $t \mapsto u(t; s, x) \in Y_t$ of (2.21). In addition, for $s_n \to s$ and $Y_{s_n} \ni x_n \to x \in Y_s$ we have $\tilde{u}(t; s_n, X_n) \to \tilde{u}(t; s, x)$ uniformly for t in compact intervals in $\mathbb{R}$, where we set $\tilde{u}(t; s, x) := u(t; s, x)$ for $t \geq s$ and $\tilde{u}(t; s, x) := x$ for $t < s$.*

The solution of the autonomous Cauchy problem is given by a strongly continuous semigroup. For the non-autonomous case this concept is generalized in the following definition.

Definition 2.3.18 *A family of bounded operators $(U(t, s))_{t,s\in\mathbb{R},t\geq s}$ on a Banach space X is called a (strongly continuous) evolution family if*

(*i*) *$U(t, s) = U(t, r)U(r, s)$ and U(s,s)=I for $t \geq r \geq s$ and $t, r, s \in \mathbb{R}$*

and

(*ii*) *the mapping $\{(\tau, \sigma) \in \mathbb{R}^2 : \tau \geq \sigma\} \ni (t, s) \mapsto U(s, t)$ is strongly continuous.*

We say that $(U(t, s))_{t,s\in\mathbb{R},t\geq s}$ solves the Cauchy problem (2.21) (on spaces Y_s) if there are dense subspaces Y_s, $s \in \mathbb{R}$, of X such that $U(t, s)Y_s \subset Y_t \subset D(\mathbf{A}(t))$ for $t \geq s$ and the function $t \mapsto U(t, s)x$ is a solution of (2.21) for $s \in \mathbb{R}$ and $x \in Y_s$.

Evolution families are also called evolution systems, evolution operators, evolution processes, propagators, or fundamental solutions. Notice that a strongly continuous semigroup $(T(t))_{t\geq 0}$ gives rise to the evolution family $U(t, s) := T(t - s)$.
For partial differential equations in which the coefficients are time-invariant, the evolution operator is just the semigroup generated by the differential operator and the corresponding boundary conditions.

2.3.3 Infinite-dimensional control theory

Before we list some results from semigroup theory applied to infinite-dimensional control theory, we first give some definitions and some standard results, see e.g. [61].
In the following, let $\mathcal{H}$ and $\mathcal{U}$ be Hilbert spaces.

Definition 2.3.19 *A function $x(.) : [t_0, t_f] \to \mathcal{H}$ is strongly measurable if $x(.)$ is the limit almost everywhere of a sequence of countably valued functions. $x(.)$ is weakly measurable if $\langle y, x(.)\rangle_{\mathcal{H}}$ is Lebesgue measurable for each $y \in \mathcal{H}$.*

Definition 2.3.20 *An operator-valued function* $\mathbf{B}(.) : [t_0, t_f] \to \mathcal{L}(\mathcal{U}, \mathcal{H})$ *is called strongly measurable if* $\mathbf{B}(.)x$ *is strongly measurable for each* $x \in \mathcal{H}$*. The set of all such functions* $\mathbf{B}(.)$ *for which* $\|\mathbf{B}(.)\|$ *is essentially bounded on* $[t_0, t_f]$ *is denoted by* $\mathcal{B}_\infty(t_0, t_f; \mathcal{U}, \mathcal{H})$

Proposition 2.3.21 $\mathcal{B}_\infty(t_0, t_f; \mathcal{U}, \mathcal{H})$ *is a Banach space together with the norm* $\|\mathbf{B}(.)\|_{\mathcal{B}_\infty} := \text{ess sup} \|\mathbf{B}(.)\|$ *and* $\mathcal{B}_\infty(t_0, t_f; \mathcal{H}, \mathcal{H})$ *is a Banach algebra.*

In [39] Curtain and Pritchard consider the linear-quadratic control problem for systems defined by integral equations given in terms of evolution families. They consider a more general class of evolution families than the ones we have reviewed here. They are called the *mild* evolution families. Unlike a strong evolution family, here just weak continuity is assumed. They show that if $U(t, s)$ is a mild evolution family, then the optimal control problem leads to an integral Riccati equation. Then, in order to obtain a differential version of the Riccati equation another type of evolution family is introduced: the *quasi* evolution family. However, to ensure uniqueness it is necessary to suppose that $U(t, s)$ is a strongly continuous evolution family. In the following we cite here the definitions of mild and quasi evolution families as well as the theorems which ensure existence and uniqueness of the differential operator Riccati equation.

Definition 2.3.22 *Let* $\mathcal{H}$ *be a real Hilbert space and* $[0, T]$ *an interval of the real line and*

$$\Delta(T) = \{(t, s) : 0 \leq s < t \leq T\}.$$

$U(., .) : \Delta(T) \to \mathcal{L}(\mathcal{H})$ *is a mild evolution family if*

$$U(t, r)U(r, s) = U(t, s) \text{ for } 0 \leq s \leq r \leq t \leq T,$$
$$U(t, s) \text{ is weakly continuous in } s \text{ on } [0, t] \text{ and in } t \text{ on } [s, T].$$

Theorem 2.3.23 *If* $U(., .)$ *is a mild evolution family on* $\Delta(T)$ *further let* $\mathbf{D} \in \mathcal{B}_\infty(0, T; \mathcal{H}, \mathcal{H})$*, then the following operator integral equation has a unique solution* $U_{\mathbf{D}}(., .)$*,*

$$U_{\mathbf{D}}(t, s)x = U(t, s)x + \int_s^t U(t, r)\mathbf{D}(r)U_{\mathbf{D}}(r, s)xdr \tag{2.22}$$

in the class of weakly continuous bounded linear operators on $\mathcal{H}$*.* $U_{\mathbf{D}}(., .)$ *is a mild evolution family and we call it the perturbed mild evolution family corresponding to the perturbation* $\mathbf{D}$*. Furthermore, if*

$$\underset{t\in[0,T]}{\text{ess sup}} \|\mathbf{D}(t)\| \leq M_1, \quad \underset{\Delta(T)}{\text{ess sup}} \|U(t, s)\| \leq M_2,$$

we have

$$\|U_{\mathbf{D}}(t, s)\| \leq M_1 \exp M_1 M_2 (t - s).$$

The integral in (2.22), as well as the ones in the following, are Bochner integrals. The Bochner integral is an extension of the Lebesgue integral to vector-valued

functions.
We recall that a function $u(.) : [a, b] \to \mathcal{U}$ is Bochner integrable if and only if $u(.)$ is strongly measurable and $\int_a^b \|u(t)\|\, dt < \infty$. For details of the Bochner integral, see for instance [61].

Definition 2.3.24 *A quasi evolution family is a mild evolution family $U : \Delta(T) \to \mathcal{H}$ such that there exists a nonzero $x \in \mathcal{H}$ and a closed linear operator $\mathbf{A}(s)$ on $\mathcal{H}$ for almost all $s \in [0, T]$ satisfying*

$$\langle y, U(t, s)x - x\rangle = \int_s^t \langle y, U(t, \rho)\mathbf{A}(\rho)x\rangle d\rho \quad \forall y \in \mathcal{H}. \tag{2.23}$$

The set of $x \in \mathcal{H}$ for which (2.23) is valid is denoted by $\mathcal{D}_{\mathbf{A}}$, and $\mathbf{A}(.)$ is called the generator of $U(., .)$.
An immediate consequence of the definition is

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

The infinite-dimensional control system considered is:

$$x(t) = U(t, s)x(s) + \int_{t_0}^t U(t, \nu)\mathbf{B}(\nu)u(\nu)d\nu, \quad 0 \le t_0 \le s \le t \le T < \infty, \tag{2.24}$$

where $U(., .)$ is a mild evolution family on the real Hilbert space $\mathcal{H}$, $u \in L^2(0, T; \mathcal{U})$, where $\mathcal{U}$ is a real Hilbert space, $x_0 \in \mathcal{H}$, and $\mathbf{B} \in \mathcal{B}_\infty(0, T; \mathcal{H}, \mathcal{H})$. With the cost functional

$$\mathcal{J}(u; t_0, x_0) = \int_{t_0}^T (\langle x(s), \mathbf{Q}(s)x(s)\rangle + \langle u(s), \mathbf{R}u(s)\rangle)ds + \langle x(T), \mathbf{G}x(T)\rangle,$$

where $x(t)$ is given by (2.24), $\mathbf{G} \in \mathcal{L}(\mathcal{H})$ is self-adjoint and nonnegative, $\mathbf{Q} \in \mathcal{B}_\infty(0, T; \mathcal{H}, \mathcal{H}))$, $\mathbf{R} \in \mathcal{B}_\infty(0, T; \mathcal{U}, \mathcal{U}))$ and for each t, $\mathbf{Q}(t)$, $\mathbf{R}(t)$ are nonnegative and self-adjoint and $\mathbf{R}(t)$ satisfies

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

Then the quadratic cost problem is:

$$\begin{array}{c}\text{Find the optimal control } u_* \in L^2(0, T; \mathcal{U}) \\ \text{which minimizes } \mathcal{J}(u; t_0, z_0).\end{array} \tag{CP}$$

The solution to (CP) is given by the following result.

Theorem 2.3.25 *The optimal control which minimizes $\mathcal{J}(u; t_0, z_0)$ is the feedback control*

$$u_*(t) = -\mathbf{R}^{-1}(t)\mathbf{B}^*(t)\mathbf{\Pi}(t)x(t), \tag{2.25}$$

where $\mathbf{\Pi}(t) \in \mathcal{B}_\infty(0,T;\mathcal{H},\mathcal{H}))$ *is a self-adjoint operator which satisfies the integral equation*

$$\begin{aligned}\mathbf{\Pi}(t)y &= U_\infty^*(T,t)\mathbf{G}U_\infty(T,t)y \\ &\quad + \int_t^T U_\infty(s,t)[\mathbf{Q}(s)+\mathbf{\Pi}(s)\mathbf{B}(s)\mathbf{R}^{-1}(s)\mathbf{B}^*(s)\mathbf{\Pi}(s)]U_\infty(s,t)yds,\end{aligned} \tag{2.26}$$

where $U_\infty(t,s)$ *is the perturbed mild family corresponding to the perturbation of* $U(t,s)$ *by* $-\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^*(t)\mathbf{\Pi}(t)$.

In Chapter 3 we refer to (2.26) as the *Riccati integral equation of Curtain and Pritchard*.

Remark 2.3.26 *An analogous result to Theorem 2.3.25 was shown by Gibson, [52, Thm 3.2]. There the optimal control is defined as (2.25) and* $\mathbf{\Pi}(t) \in \mathcal{B}_\infty(0,T;\mathcal{H},\mathcal{H}))$ *is a self-adjoint operator which satisfies*

$$\begin{aligned}\mathbf{\Pi}(t)y &= U^*(T,t)\mathbf{G}U(T,t)y \\ &\quad + \int_t^T U(s,t)[\mathbf{Q}(s)-\mathbf{\Pi}(s)\mathbf{B}(s)\mathbf{R}^{-1}(s)\mathbf{B}^*(s)\mathbf{\Pi}(s)]U(s,t)yds,\end{aligned} \tag{2.27}$$

where $U(s,t)$ *is a strong evolution family. Gibson showed that if* $\mathbf{\Pi}(t)$ *is the unique solution of (2.27), then it is also the unique solution of (2.26). He called (2.27) as the first Riccati integral equation.*

Theorem 2.3.27 *Let* $U(t,s)$ *be a quasi evolution family on* $\mathcal{H}$. *Then the solution of the integral equation (2.26) satisfies the following inner product differentiated Riccati equation:*

$$\begin{aligned} &\tfrac{d}{dt}\langle\mathbf{\Pi}(t)z,y\rangle + \langle\mathbf{\Pi}(t)z,\mathbf{A}(t)y\rangle + \langle\mathbf{A}(t)z,\mathbf{\Pi}(t)y\rangle \\ &-\langle\mathbf{\Pi}(t)\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^*(t)\mathbf{\Pi}(t)z,y\rangle + \langle\mathbf{Q}(t)z,y\rangle = 0 \quad \text{a.e. on } [t_0,T], \\ &\mathbf{\Pi}(T) = \mathbf{G} \qquad\qquad\qquad\qquad\quad \text{for } z,y \in \mathcal{D}_\mathbf{A}.\end{aligned} \tag{2.28}$$

If $\mathbf{B}$, $\mathbf{Q}$ *and* $\mathbf{R}$ *are strongly continuous on* $[0,T]$, *then (2.28) is satisfied everywhere on* $[t_0,T]$.

Theorem 2.3.28 *Let* $U(t,s)$ *be a strong evolution family with generator* $\mathbf{A}(t)$ *such that* $\langle U(t,r)\mathbf{A}(r)z,y\rangle$ *is integrable with respect to* r *on* (s,t) *for all* $y \in \mathcal{H}$ *and* $z \in \mathcal{D}_\mathbf{A}$. *If* $\bar{\mathcal{D}}_\mathbf{A} = \mathcal{H}$, *then (2.28) has a unique solution in the class of self-adjoint weakly continuous operators* $\mathbf{\Pi}(.)$, *such that* $\langle z,\mathbf{\Pi}(.)y\rangle$ *is absolutely continuous for all* $z,y \in \mathcal{D}_\mathbf{A}$.

Gibson [52], considers a strongly continuous evolution family. However, the results concerning optimal control and the Riccati integral equations hold if weak continuity is assumed (i.e., if a mild evolution family is assumed) and $\mathcal{H}$ is separable. Basically strong continuity or weak continuity and separability of $\mathcal{H}$ are needed to guarantee strong measurability of $U(.,.)$ in either argument. Since strong measurability of $U(.,.)$ implies only weak measurability of $U^*(.,.)$, strong measurability of $U^*(.,.)$ in either argument is required also.

Referring to the optimal control problem (CP), suppose that $\{U_i(.,.)\}$ is a sequence of evolution operators on $\mathcal{H}$ and that $\{\mathbf{B}_i(.)\}$, $\{\mathbf{Q}_i(.)\}$, $\{\mathbf{R}_i(.)\}$, and $\{\mathbf{G}_i\}$ are sequences of operators in $\mathcal{B}_\infty(t_0,T;\mathcal{U},\mathcal{H})$, $\mathcal{B}_\infty(t_0,T;\mathcal{H},\mathcal{H})$, $\mathcal{B}_\infty(t_0,T;\mathcal{U},\mathcal{U})$ and $\mathcal{L}(\mathcal{H})$, respectively, with $\mathbf{Q}_i(.)$, $\mathbf{R}_i(.)$, and $\mathbf{G}_i$ nonnegative and self-adjoint. We consider the sequences of optimal control problems corresponding to these sequences of operators. Suppose that, for each $x \in \mathcal{H}$ and $u \in \mathcal{U}$,

$$\begin{array}{lllll}
\text{(i)} & U_i(t,s)x \to U(t,s)x & \text{strongly,} & t_0 \le s \le t \le T, \\
\text{(ii)} & U_i^*(t,s)x \to U^*(t,s)x & \text{strongly,} & t_0 \le s \le t \le T, \\
\text{(iii)} & \mathbf{B}_i(t)u \to \mathbf{B}(t)u & \text{strongly a.e.,} & \\
\text{(iv)} & \mathbf{B}_i^*(t)x \to \mathbf{B}^*(t)x & \text{strongly a.e.,} & \\
\text{(v)} & \mathbf{Q}_i(t)x \to \mathbf{Q}(t)x & \text{strongly a.e.,} & \\
\text{(vi)} & \mathbf{R}_i(t)u \to \mathbf{R}(t)u & \text{strongly a.e.,} & \\
\text{(vii)} & \mathbf{G}_i x \to \mathbf{G}x & \text{strongly,} &
\end{array} \tag{G}$$

as $i \to \infty$. We require $\|U_i(t,s)\|$, $\|\mathbf{B}_i\|_{\mathcal{B}_\infty}$, $\|\mathbf{Q}_i\|_{\mathcal{B}_\infty}$, $\|\mathbf{R}_i\|_{\mathcal{B}_\infty}$ and $\|\mathbf{G}_i\|$ to be uniformly bounded in i, t, and s and require a constant m such that for each i, $\mathbf{Q}_i(t) \ge m > 0$ for almost all t.

Theorem 2.3.29 *Let (G) hold, along with the uniform bounds. For our sequence of control problems, denote the initial states by $x_i(t_0)$, and let $x_i(t_0) \to x(t_0)$; denote the optimal controls by $u_i(.)$, the optimal trajectories by $x_i(.)$, and the solutions of the Riccati integral equations by $\mathbf{\Pi}_i(.)$. For the problem (CP), denote the corresponding quantities by $x(t_0)$, $u(.)$, $x(.)$, and $\mathbf{\Pi}(.)$. Then we have*

$$\begin{array}{ll}
u_i(t) \to u(t) & \text{strongly a.e. and in } L^2(t_0,T;\mathcal{U}), \\
x_i(t) \to x(t) & \text{strongly pointwise and in } L^2(t_0,T;\mathcal{H})
\end{array} \tag{2.29}$$

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

$$\mathbf{\Pi}_i(t)x \to \mathbf{\Pi}(t)x \quad \text{strongly pointwise and in } L^2(t_0,T;\mathcal{H}). \tag{2.30}$$

If $U(.,.)$ is strongly continuous and $\mathbf{B}(.)$, $\mathbf{B}^*(.)$, $\mathbf{Q}(.)$, and $\mathbf{R}(.)$ are piecewise strongly continuous, uniform convergence in (G) implies uniform convergence in (2.29)–(2.30).

CHAPTER

THREE

Convergence theory

If we semi-discretize an infinite-dimensional linear-quadratic regulator (LQR) problem in space, then we obtain a finite-dimensional LQR problem. In this chapter, 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. First, we will give a brief survey about the theoretical background of LQR problems in Section 3.1. Then, in Section 3.2 we state the infinite-dimensional LQR problem for which an existence and uniqueness theorem is presented. After that, in Section 3.3 we consider a family of finite-dimensional LQR problems defined on subsets of the original state space. Then, in section 3.4 we show an approximation theorem which gives us a theoretical justification for the numerical method used for the linear problems described in this thesis. Finally, in Section 3.5 we extend our result for the non-autonomous case, i.e., the case in which the system dynamics is modeled by partial differential equations with time-varying coefficients.

3.1 Introduction

The linear-quadratic control problem for finite-dimensional systems is a well understood subject, its theory can be found in many textbooks see e.g. [5, 8, 35, 106, 117]. A generalization of the finite-dimensional theory has been developed for infinite-dimensional systems, see e.g. [26, 27, 41, 75, 76, 77]. Many control, stabilization and parameter identification problems can be reduced to the linear-quadratic regulator (LQR) problem. Particularly, the LQR problem for parabolic systems has been studied in detail in the past 30 years. The classical reference is the book of Lions [82], there he presented a complete solution for evolution equations of parabolic type on both finite and infinite-time intervals. His variational approach leads to a Hamiltonian system of equations, which is then synthesized to obtain Riccati equations. This allows, relatively easy, to extend the problem to hyperbolic and other classes of partial differential equa-

tions as well as boundary control and point observations [83].
In the literature, many authors have considered the linear-quadratic control problem for infinite-dimensional systems in the context of semigroup theory. The first attempt to develop a general semigroup framework for solving quadratic control problems with unbounded input and output operators was done by Pritchard and Salamon [99]. This can be seen as an abstract version of Lions's work because the results applies both to parabolic and hyperbolic systems as well as retarded and neutral functional differential equations. Depending on the conditions imposed on the semigroups related to the dynamics, the solutions of control problems lead to infinite-dimensional integral Riccati equations or differential Riccati equations for the finite-time horizon case and to infinite-dimensional algebraic Riccati equations for the infinite-time horizon case. Another approach was adopted by Datko who solved the problem on finite and infinite-time interval without introducing a Riccati equation, see [42, 43]. The theory of quadratic cost optimal control for infinite-dimensional systems can be found in many books, e.g [41, 81], in particular the books of Bensoussan et al. [26, 27] cover the subject in detail. An excellent survey of the most recent results, as well as numerical aspects, can be found in the books by Lasiecka and Triggiani [76, 77].
Approximation schemes for Riccati equations in infinite-dimensional spaces have been proposed in the recent years. Chronologically, the first reference is Gibson [52], who presented an approximation technique to reduce the inherently infinite-dimensional problems to finite-dimensional analogues in terms of the Riccati integral equations. However, in order to make comparisons with finite-dimensional theory and for computational applications (which is one objective of this thesis), infinite-dimensional differential Riccati equations have to be considered. The result proposed by Gibson requires the approximating problems to be defined on the entire original state space, this leads to tedious technical considerations. Assuming that the dynamics is modeled by an analytic semigroup, Banks and Kunisch [12] avoid these technical considerations for the infinite-time horizon case. An extension of this result for boundary control problems was given by Benner and Saak in [24]. For the infinite-time horizon case convergence rates for some type of problems have been proved by Lasiecka and Triggiani [76, 77].
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 proofs here follow mostly from the abstract theory develop by Gibson [52], and from the ideas for the infinite-time horizon case presented in [12, 24].

3.2 Infinite-dimensional systems

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, $\mathbf{A}$: $\mathrm{dom}(\mathbf{A}) \subset \mathcal{H} \to \mathcal{H}$ 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

$$\begin{array}{ll} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), & t > 0, \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), & t > 0, \\ \mathbf{x}(0) = \mathbf{x_0}, & \end{array} \tag{3.1}$$

and a cost functional

$$J(\mathbf{u}) := \int_0^{T_f} \langle \mathbf{x}, \mathbf{Q}\mathbf{x} \rangle_{\mathcal{H}} + \langle \mathbf{u}, \mathbf{R}\mathbf{u} \rangle_{\mathcal{U}} \, dt + \langle \mathbf{x}_{T_f}, \mathbf{G}\mathbf{x}_{T_f} \rangle_{\mathcal{H}}, \tag{3.2}$$

where we assume that (3.1) has a unique solution. 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 $\mathbf{x}(., T_f)$. The abstract linear optimal regulator problem can then be stated as

$$\begin{array}{c} \text{Minimize } J(\mathbf{u}) \text{ over } L^2(0, T_f; \mathcal{U}) \\ \text{subject to } \mathbf{x} = \mathbf{x}(.; \mathbf{u}) \text{ satisfying (3.1).} \end{array} \tag{$\mathcal{R}$}$$

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 $J(x_0, \mathbf{u})$ is finite. We now have to consider the operator differential Riccati equation:

$$\begin{array}{lcl} \dot{\mathbf{\Pi}}(t) & = & -(\mathbf{Q} + \mathbf{A}^*\mathbf{\Pi}(t) + \mathbf{\Pi}(t)\mathbf{A} - \mathbf{\Pi}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{\Pi}(t)), \\ \mathbf{\Pi}(T_f) & = & \mathbf{G}. \end{array} \tag{3.3}$$

We define a solution of (3.3) in the interval $[0, T_f]$ as an operator $\mathbf{\Pi}(t)$ such that $\mathbf{\Pi}(T_f) = \mathbf{G}$ and for all φ, $\psi \in \text{dom(A)}$, $\langle \varphi, \mathbf{\Pi}(.)\psi \rangle$ is differentiable in $[0, T_f]$ and satisfies the equation,

$$\begin{array}{lcl} \frac{d}{dt}\langle \varphi, \mathbf{\Pi}(t)\psi \rangle & = & -(\langle \varphi, \mathbf{Q}\psi \rangle + \langle \mathbf{A}\varphi, \mathbf{\Pi}(t)\psi \rangle + \langle \mathbf{\Pi}(t)\varphi, \mathbf{A}\psi \rangle \\ & & -\langle \mathbf{\Pi}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{\Pi}(t)\varphi, \psi \rangle) \end{array} \tag{3.4}$$

as is defined in [26, Def. 2.1, pp. 142].

Theorem 3.2.1 *The unique control which minimizes (3.2) is the linear* feedback *control,*

$$\mathbf{u}_*(t) = -\mathbf{R}^{-1}\mathbf{B}^*\mathbf{\Pi}(t)\mathbf{x}_*(t),$$

where $\mathbf{\Pi}(t)$ is the unique nonnegative self-adjoint solution of (3.3). The corresponding optimal trajectory is given by

$$\mathbf{x}_* = S(t)\mathbf{x}_0,$$

where $S(t)$ is the strongly continuous semigroup generated by $\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{\Pi}(t)$. The minimum value of the cost functional is $(\mathbf{\Pi}(0), \mathbf{x}_0)$.

Proof. The proof of this theorem is given, e.g., in [76, 41].

□

Remark 3.2.2 *Note that any solution of (3.3) is self-adjoint, and that $\mathbf{\Pi}(.)$ is nonnegative if $\mathbf{G}$ is.*

3.3 Approximation by finite-dimensional systems

In order to solve $(\mathcal{R})$ for practical problems, we have to find suitable finite-dimension approximations to the solutions given in Theorem 3.2.1.
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 $B^N \in \mathcal{L}(U, \mathcal{H}^N)$, 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:

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

with respect to

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

$(\mathcal{R}^N)$ is a linear regulator problem in the finite-dimensional state space $\mathcal{H}^N$. If $Q^N \geq 0$, $\mathbf{R} > 0$ then, by Theorem 2.2.6, 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:

$$\begin{array}{l} \dot{\Pi}^N(t) = -(Q^N + A^{N*}\Pi^N(t) + \Pi^N(t)A^N - \Pi^N(t)B^N\mathbf{R}^{-1}B^{N*}\Pi^N(t)), \\ \Pi^N(t_f) = G^N, \end{array} \qquad (3.5)$$

and $x_*^N(t)$ is the corresponding solution of the state equation with $\mathbf{u}(t) = u_*(t)^N$.
Let us now consider a related family of regulator problems, in which the operators are defined in the whole space,

Minimize:

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

with respect to

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

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 $\bar{\mathbf{G}}^N x^N(t_f) = G^N x^N(t_f)$.

Applying Theorem 3.2.1 the optimal control is given in terms of the solution of

$$\begin{aligned}&\dot{\bar{\mathbf{\Pi}}}^N(t) = -(\bar{\mathbf{Q}}^N + \bar{\mathbf{A}}^{N*}\bar{\mathbf{\Pi}}^N(t) + \bar{\mathbf{\Pi}}^N(t)\bar{\mathbf{A}}^N - \bar{\mathbf{\Pi}}^N(t)B^N\mathbf{R}^{-1}B^{N*}\bar{\mathbf{\Pi}}^N(t)),\\ &\bar{\mathbf{\Pi}}^N(t_f) = \bar{\mathbf{G}}^N.\end{aligned} \tag{3.6}$$

Note that

$$\bar{\mathbf{\Pi}}^N(t) = \Pi^N(t)P^N. \tag{3.7}$$

In fact, if in (3.5) we replace Q^N, A^N, G^N by Q^NP^N, A^NP^N, G^NP^N, respectively, then it can be considered as an equation on $\mathcal{H}$. Moreover, (3.6) and (3.5) 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 (3.7) holds.

3.4 Convergence statement

The main result of this chapter, Theorem 3.4.1, is essentially contained in [52]. The difference here, similar to [12, 25], is that each of the finite-dimensional approximation problems are defined in a subspace of the state space, whereas in [52], the approximation problems have to be defined in the entire state space. That is, the result is formulated using $(\mathcal{R}^N)$ rather than $(\bar{\mathcal{R}}^N)$. This avoids some technical difficulties, see [12].
We will assume, similar to [12, (H2)],

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

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 3.4.1 *Let* (H) *hold, then*

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

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

$$\Pi^N(t)P^N\varphi \to \mathbf{\Pi}(t)\varphi \quad \text{uniformly in } t \in [0, T_f]. \tag{3.8}$$

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

Proof. Let $\mathbf{\Pi}(t)$ be the unique element of $\mathcal{B}_\infty(0,T_f;\mathcal{H},\mathcal{H})$, see Definition 2.3.20, which satisfies the first Riccati integral equation (see Remark 2.3.26). By calculations in [52, pp. 544-546], $\mathbf{\Pi}(t)$ is also the unique solution of the Riccati integral equation of Curtain and Pritchard [39], equation (2.26). Theorems 2.3.28 and 2.3.27 ensure that $\mathbf{\Pi}(t)$ uniquely satisfies the infinite-dimensional differential Riccati equation (3.4). Let $\bar{\mathbf{\Pi}}^N(t)$ be the Riccati operator related to the problem $(\bar{\mathcal{R}}^N)$. By (3.7) the theorem holds as a direct consequence of Theorem 2.3.29.

□

We point out that is it possible to prove an analogue to Theorem 3.4.1 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 modifying hypotheses (H) like,

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

we can state, similar to Theorem 3.4.1,

Theorem 3.4.2 *Let* (H′) *hold, then*

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

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

$$\|\Pi^N(t)P^N\varphi - P^N\mathbf{\Pi}(t)\varphi\|_N \to 0 \quad \text{uniformly in } t \in [0,T_f]. \tag{3.9}$$

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

Proof. The proof follows very close to the one of Theorem 3.4.1 once an analogue to Theorem 2.3.29, which permits $\mathcal{H}^N \nsubseteq \mathcal{H}$, has been proven.

□

Note that the lemma which the proof of Theorem 2.3.29 relies, [52, Lemma 5.1, p. 560], can be modified as:

Lemma 3.4.3 *Let X be a Banach space, let $\{X^N\}_{N\geq 2}$ be a sequence of Banach spaces and let $P^N : \mathcal{H} \to \mathcal{H}^N$ 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, $A_N(\cdot)P^N g_N(\cdot)$ converge 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 3.4.3, and Lemma 5.1 [52, p. 560] let us prove an analogue to Theorem 2.3.29 (Theorem 3.5.2, next section), 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. In Chapter 6, Section 6.1, we use a finite element Galerkin approximation which fits the requirements of Theorem 3.4.1.

Remark 3.4.4 *The theoretical results proved in this chapter give us an approximation framework for computation of Riccati operators that can be guaranteed to converge to the Riccati operator required in feedback control problems.*
A similar result for nonlinear problems is an open problem. However, in this case model predictive control technics can be applied [18, 68]. There the equation is linearized and linear problems have to be solved on subintervals of $[0, T_f]$. In Chapter 6, Section 6.2, we present numerical examples using this technique.

3.5 The non-autonomous case

We consider now partial differential equations in which the coefficients are time-varying. Then, the system dynamics is modeled by an evolution operator. In the following we will see that the approximation results presented in the previous section (Theorems 3.4.1, 3.4.2) can be extended to this case.
Let $\mathcal{H}$ and $\mathcal{U}$ be real Hilbert spaces and consider an evolution process defined by

$$x(t) = U(t,s)x(s) + \int_0^t U(t,\nu)\mathbf{B}(\nu)u(\nu)d\nu, \tag{3.10}$$

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 (3.10) 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 as in Definition 2.3.24. We use the integral form of (3.10) in our presentation to closely follow [39, 52].
With 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 (3.10), $\mathbf{G} \in \mathcal{L}(\mathcal{H})$ 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:

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

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 projection. 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 [52]. Specifically, in (G') the evolution operators corresponding to the approximating problems are defined on a subspace of the original state space of the original problem, whereas in [52] 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 3.5.1 *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*

$$\begin{array}{ll}
u^N(t) \to u(t) & \text{strongly a.e. and in } L^2(0,T_f;\mathcal{U}), \\
x^N(t) \to x(t) & \text{strongly pointwise and in } L^2(0,T_f;\mathcal{H}),
\end{array} \tag{3.11}$$

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

$$\Pi^N(t)P^N\varphi \to \mathbf{\Pi}(t)\varphi \quad \text{strongly pointwise and in } L^2(0,T_f;\mathcal{H}). \tag{3.12}$$

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 (3.11)–(3.12).*

Proof. As for the autonomous case the sequence of control problems are defined on 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 ($\bar{\mathcal{R}}^N$). Thus, assuming similar arguments, on $\mathbf{\Pi}(\mathbf{t})$, to the ones in the proof of Theorem 3.4.1, the proof of Theorem 3.5.1 follows directly from Theorem 2.3.29.

□

Like in the autonomous case (Theorem 3.4.2), it is possible to prove an analogue to Theorem 3.5.1 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. If we modify (G') like,

(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 constants 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 $\left\|B^{N*}P^N\varphi - \mathbf{B}^*\varphi\right\|_U \to 0$.
(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.

(GN')

We can state, similar to Theorem 3.4.2.

Theorem 3.5.2 *Under the hypotheses of Theorem 3.5.1 with (GN') instead of* (G'), *we have*

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

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

$$\Pi^N(t)P^N\varphi \to \mathbf{\Pi}(t)\varphi \quad \text{uniformly on } [0, T_f]. \tag{3.13}$$

Here u^N, x^N, denote the optimal control and trajectories, respectively, for our sequence of control problems, and u and x for the $(\mathcal{NAR})$.

Proof. As we point out in last section, the proof follows as a consequence of the repeated application of Lemma 3.4.3 and Lemma 5.1 [52, p. 560].

□

Remark 3.5.3 *The results proposed in this section will be particularly useful solving nonlinear problems in model predictive control and receding horizon*

context in Chapter 6 Section 6.3. 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.

CHAPTER

FOUR

Numerical methods for DREs

In this chapter we study the numerical solution of DREs arising in optimal control problems for parabolic PDEs. First, in Section 4.1 we review the existing methods for solving DREs and discuss whether these methods are suitable for large-scale computations. In Section 4.2 we suggest an efficient implementation for the backward differentiation formulae (BDF) methods based on a low rank version of the alternating direction implicit (ADI) iteration. After that, in Section 4.3, we study the Rosenbrock methods and propose an efficient algorithm for solving large-scale DREs based also on the ADI iteration. Finally, in Section 4.4 a new method for determining sets of shift parameters for the ADI iteration is described which improves its efficiency.
As we point out in Chapter 3, solving nonlinear control problems in model predictive control context will lead us to solve DREs with time-varying coefficients. Hence, throughout this chapter we consider time-varying symmetric DREs of the form

$$\begin{aligned} \dot{X}(t) &= Q(t) + X(t)A(t) + A^T(t)X(t) - X(t)S(t)X(t), \\ X(t_0) &= X_0, \end{aligned} \tag{4.1}$$

where $t \in [t_0, t_f]$ and $Q(t)$, $A(t)$, $S(t)$, $\in \mathbb{R}^{n\times n}$ are piecewise continuous locally bounded matrix-valued functions. 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.

4.1 Known methods

The numerical methods for DREs of the form (4.1) can essentially be distinguished into five classes. Note that the solution matrix of the DRE is a symmetric $n \times n$ matrix. Even in case symmetry is exploited, the storage needed is of size $n(n+1)/2$. For example, for a semi-discretized 2D PDE problem with

say, 11,000 degrees of freedom, this would require about 500 MB of storage for each time step if double precision is to be used! Therefore, we will examine the available methods regarding their potential to circumvent the storage of $X(t)$ as a square matrix. This section is essentially contained in [21].

The naive approach. The first idea is to vectorize the DRE, i.e., to unroll the matrices into vectors and to integrate the resulting system of n^2 differential equations using any kind of numerical integration scheme. This approach is not suitable for large-scale problems, as for implicit methods, nonlinear systems of equations with n^2 unknowns have to be solved in each time step. This can be reduced exploiting symmetry to $n(n+1)/2$, but still this would require $\mathcal{O}(n^2)$ workspace, [72, 86].

Linearization. The second type of methods is based on transforming the quadratic DRE into the system of linear first-order matrix differential equations

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} U(t) \\ V(t) \end{bmatrix} &= \underbrace{\begin{bmatrix} -A(t) & S(t) \\ Q(t) & A(t)^T \end{bmatrix}}_{:=H(t)} \begin{bmatrix} U(t) \\ V(t) \end{bmatrix}, \qquad t \in (t_0, T], \\ \begin{bmatrix} U(t_0) \\ V(t_0) \end{bmatrix} &= \begin{bmatrix} U_0 \\ V_0 \end{bmatrix}, \end{aligned} \tag{4.2}$$

where $U(t) \in \mathbb{R}^{n\times n}$, $V(t) \in \mathbb{R}^{n\times n}$ and $V_0 U_0^{-1} = X(t_0)$ for some $U_0 \in \mathbb{R}^{n\times n}$ invertible and some $V_0 \in \mathbb{R}^{n\times n}$. If the solution of (4.1) exists on the interval $[t_0, T]$, then the solution of (4.2) exists, $U(t)$ is invertible on $[t_0, T]$, and

$$X(t) = V(t)U^{-1}(t). \tag{4.3}$$

Conversely, if the solution of (4.2) exists and $U(t)$ is nonsingular for all $t \in [t_0, T]$, then the solution of (4.1) exists in the same interval and is given by (4.3). The linear differential equation (4.2) is a Hamiltonian differential equation. In the time-invariant case, this allows an efficient integration for dense problems, [78], using numerical methods for the Hamiltonian eigenproblem.

Another approach which is applicable to time-varying systems uses the fundamental solution of the linear first-order ordinary differential equation. This method, called now the Davison-Maki method, is proposed in [45]. A modified variant, avoiding some numerical instabilities due to the inversion of possibly ill-conditioned matrices, is proposed in [70]. The exponential of the $2n \times 2n$-matrix $H(t_0)$ is required. The application of this method for large-scale problems might be investigated further by approximating $e^H \approx V e_k^H V^T$, $\operatorname{range}(V) = \operatorname{span}\{x, Hx, \ldots, H^{k-1}x\}$, $k \ll n$.

Chandrasekhar's method. The third type of algorithms is applicable to symmetric time-invariant DREs and is based on the transformation of (4.1)

into two coupled systems of nonlinear differential equations, the so-called *Chandrasekhar system*

$$\begin{aligned} \dot{L} &= (K^T G^T - A^T)L, & L(0) = C \in \mathbb{R}^{n\times l}, \\ \dot{K} &= -G^T L L^T, & K(0) = G^T X_0 \in \mathbb{R}^{m\times n}, \end{aligned} \tag{4.4}$$

where $Q(t) \equiv CC^T$, $S(t) \equiv GG^T$.
The relationship between L, K, and X is given by

$$\begin{aligned} K(t) &= G^T X(t), \\ L(t)L^T(t) &= -\dot{X}(t), \\ X(t)A + A^T X(t) &= K^T(t)K(t) + L(t)L^T(t) - CC^T, \end{aligned} \tag{4.5}$$

and therefore, the solution of the DRE can be recovered from that of (4.4). The method can be adapted to the time-varying case, see [73], but there are several numerical difficulties involved in integrating (4.4), see [102]. In general, the method is unstable and is therefore not considered here any further although it is suitable for large-scale problems [11].

Superposition methods. This type of methods is based on the superposition property of Riccati solutions, see [59]. The general solution of a DRE can be expressed as a nonlinear combination of at most five independent solutions. This class of methods requires integration of the DRE several times with different initial conditions before applying the complex superposition formulae and the computational complexity therefore is too high to apply these formulae to the large-scale problems considered here.

Matrix-versions of standard ODE methods. These methods solve the DRE using matrix-valued algorithms based on standard numerical algorithms (see [37, 46]) for solving ordinary differential equations (ODEs). As we are concerned with stiffness, we only consider implicit methods here. In order to use the given structure as much as possible, we are interested in methods which, written in matrix form, yield an algebraic Riccati equation (ARE) as the nonlinear system of equations to be solved in each time step. It turns out that there is a vast variety of methods that are applicable here, e.g., the backward differentiation formulae (BDF), the midpoint and trapezoidal rules.

The BDF schemes allow an efficient implementation for the large-scale problems considered here. Moreover, BDF schemes are particularly suitable for stiff ODEs. Therefore, we will concentrate on this class of methods in Section 4.2.
Diagonally implicit Runge-Kutta (DIRK) methods or collocation methods offer an alternative to the BDF methods for stiff problems. In particular, linearly implicit one-step methods (better known as Rosenbrock methods) give satisfactory results see, e.g., [31, 58]. We focus on the Rosenbrock methods in Section 4.3. The application of these methods to the DRE implies the solution of one Lyapunov equation in each stage of the method.

We solve the resulting Lyapunov equation exploiting the given structure of the coefficient matrices and show that a suitable implementation for large-scale problems is also feasible for these methods.

In the next section, we describe the matrix-valued implementation of BDF methods for DREs.

4.2 The backward differentiation formulae

4.2.1 Linear multistep methods

Linear multistep methods (LMM) use information from previous integration steps to construct higher-order approximations in a simple fashion. They typically come in families. The most popular family for nonstiff problems is the Adams family and the most popular for stiff problems is the backward differentiation formula (BDF) family. These families generalize the explicit and implicit Euler method, respectively. For an introduction to LMM see [57, 58].
These methods form the basis for a wide variety of ordinary differential equations (ODE) integrators, see [32, 62, 104]. Whereas they are very efficient in advancing the integration, the implementation of suitable step size selection strategies can be non-trivial.
In the following we consider the ODE system

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0, \tag{4.6}$$

across a step $t_i = t_{i-1} + h_i$, we denote $x_i \approx x(t_i)$ and $f_i := f(t_i, x_i)$. The general form of a p-step linear multistep method is given by

$$\sum_{i=0}^{p} \alpha_i x_{k-i} = h \sum_{i=0}^{p} \beta_i f_{k-j}, \tag{4.7}$$

where α_i, β_i are the coefficients of the method and h is the step size, which in general is assumed constant. Moreover it is assumed that $\alpha_0 \neq 0$, $|\alpha_i| + |\beta_i| \neq 0$, and $\alpha_0 = 1$, the latter just to eliminate arbitrary scaling. The method is called linear because (4.7) is linear in f. It is explicit if $\beta_0 = 0$ and implicit otherwise. For the general LMM (4.7) we assume the past values, (x_{k-j}, f_{k-j}), $j = 1, \ldots, p$, known in an equally spaced mesh. If at time t_{k-1} we want to take a step of size h_k, $h_k \neq h_{k-1}$, then we need the solution values at past times

$$t_{k-1} - jh_k, \quad 1 \leq j \leq p - 1. \tag{4.8}$$

To approximate these values there are three main options:

1. Compute the missing values using polynomial interpolation at the nodes from (4.8).

2. Derive a formula based on unequally spaced data.

3. Construct the polynomial interpolating x_{k-i} at the last $p+1$ values on the unequally spaced mesh. Then construct a new polynomial ψ interpolating the first polynomial at the nodes from (4.8), satisfying

$$\psi'(t_k) = f(t_k, \psi(t_k)).$$

Finally, approximate the values using this new polynomial.

For a detailed explanation see, e.g., [7].
Once that the current step has been accepted the next task is to choose the step size and order for the next step. We briefly summarize BDF methods as well as one strategy for adaptative control of order and step size for these methods in the following.

4.2.2 BDF methods

In this section we will derive fixed and variable coefficients formulae for the BDF methods. These methods are usually implemented together with a Newton iteration to solve the nonlinear algebraic equations involved at each step, see e.g., [7, 31, 58, 57]. Here, we mostly follow the book of Ascher and Petzold, [7]. Let $\phi(t)$ be the p-th degree polynomial interpolating $\{x_i \approx x(t_i)\}$ at points $\{t_k, t_{k-1}, \ldots, t_{k-p}\}$, then $\phi(t)$ can be expressed as using the Newton form as

$$\phi(t) = \sum_{j=0}^{p} \prod_{i=0}^{j-1} (t - t_{k-i})[x_k, x_{k-1}, \ldots, x_{k-j}],$$

where the divided differences are defined recursively by

$$\begin{aligned} [x_k] &= x_k, \\ [x_k, x_{k-1}, \ldots, x_{k-i}] &= \frac{[x_k, x_{k-1}, \ldots, x_{k-i+1}] - [x_{k-1}, x_{k-2}, \ldots, x_{k-i}]}{t_k - t_{k-i}}. \end{aligned}$$

The BDF methods are defined by solving an equation of the form

$$\dot{\phi}(t_k) = f(t_k, x_k).$$

Note that the mesh does not need to be equidistant here. Now differentiating $\phi(t)$ yields

$$\dot{\phi}(t) = \sum_{j=1}^{p} \left(\sum_{i=0}^{j-1} \prod_{\substack{\ell=0 \\ \ell \neq i}}^{j-1} (t - t_{k-\ell}) \right) [x_k, x_{k-1}, \ldots, x_{k-j}],$$

and evaluating $\phi(t)$ at t_k we obtain

$$\sum_{j=1}^{p} \prod_{i=1}^{j-1} (t_k - t_{k-i})[x_k, x_{k-1}, \ldots, x_{k-j}] = f(t_k, x_k). \tag{4.9}$$

p	β	α_0	α_1	α_2	α_3	α_4	α_5	α_6
1	1	1	-1					
2	$\frac{2}{3}$	1	$-\frac{4}{3}$	$\frac{1}{3}$				
3	$\frac{6}{11}$	1	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$			
4	$\frac{12}{25}$	1	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$		
5	$\frac{60}{137}$	1	$-\frac{300}{137}$	$\frac{300}{137}$	$-\frac{200}{137}$	$\frac{75}{137}$	$-\frac{12}{137}$	
6	$\frac{60}{147}$	1	$-\frac{360}{147}$	$\frac{450}{147}$	$-\frac{400}{147}$	$\frac{225}{147}$	$-\frac{72}{147}$	$\frac{10}{147}$

Table 4.1: Coefficients of the BDF k-step methods up to order 6.

On an equally spaced mesh, i.e., using a constant step size h, (4.9) yields the p-step fixed-coefficient BDF methods

$$\sum_{i=1}^{p} \frac{1}{i} \nabla^i x_k = h f(t_k, x_k),$$

where the backward differences[1] are defined recursively by,

$$\begin{aligned} \nabla^0 x_j &= x_j, \\ \nabla^i x_j &= \nabla^{i-1} x_j - \nabla^{i-1} x_{j-1}. \end{aligned}$$

This can be written as a general p-step method of the form,

$$\sum_{i=0}^{p} \alpha_i x_{k-i} = h\beta f(t_k, x_k), \tag{4.10}$$

where α_i, β are the coefficient of the method. The first six members of this family are listed in Table 4.1. These methods become unstable for $p > 6$.
Working on unequally spaced meshes, we can derive the variable-coefficient BDF by rewriting (4.9) as a general multistep method similar to (4.10),

$$\sum_{i=0}^{p} \tilde{\alpha}_i x_{k-i} = h_k \tilde{\beta} f(t_k, x_k), \tag{4.11}$$

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

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

Better stability properties are obtained with these methods. They are specially well suited for problems which require frequent or drastic changes of step size, however the Jacobian matrix of the Newton's iteration depends not only on the

[1] Note that $\nabla^p f \approx h^p f^{(p)}$.

current step but also on the sequence of the $p-1$ past steps; so it is not possible to save and reuse this matrix as is done by other methods.
To derive the variable-coefficient form of the second order BDF method from (4.9) we get

$$f(t_k, x_k) = \frac{x_k - x_{k-1}}{h_k} + \frac{h_k}{h_k + h_{k-1}}\left(\frac{x_k - x_{k-1}}{h_k} - \frac{x_{k-1} - x_{k-2}}{h_{k-1}}\right),$$

where h_k, h_{k-1}, are the step sizes. Re-arranging terms we can write the last equation as a general multistep method,

$$\tilde{\alpha}_0 x_k + \tilde{\alpha}_1 x_{k-1} + \tilde{\alpha}_2 x_{k-2} = h_k \tilde{\beta}_0 f(t_k, x_k),$$

where

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

For a third order BDF method, from (4.9) we get

$$\tilde{\alpha}_0 x_k + \tilde{\alpha}_1 x_{k-1} + \tilde{\alpha}_2 x_{k-2} + \tilde{\alpha}_3 x_{k-3} = h_k \tilde{\beta}_0 f(t_k, x_k),$$

where

$$\begin{aligned}
\tilde{\beta}_0 &= \frac{1}{\tilde{\alpha}}, \\
\tilde{\alpha}_0 &= 1, \\
\tilde{\alpha}_1 &= -\frac{1}{\tilde{\alpha}}\left[1 + \left(\frac{h_k}{h_k + h_{k-1}} + \frac{h_k}{h_k + h_{k-1} + h_{k-2}}\right)\left(1 + \frac{h_k}{h_{k-1}}\right)\right. \\
&\quad \left. + \left(\frac{h_k}{h_k + h_{k-1} + h_{k_2}}\right)\left(\frac{h_k}{h_{k-1}}\right)\left(\frac{h_k + h_{k-1}}{h_{k-1} + h_{k-2}}\right)\right], \\
\tilde{\alpha}_2 &= \frac{1}{\tilde{\alpha}}\left[\frac{h_k}{h_{k-1}}\left(\frac{h_k}{h_k + h_{k-1}} + \frac{h_k}{h_k + h_{k-1} + h_{k-2}}\right) + \right. \\
&\quad \left. \left(\frac{h_k + h_{k-1}}{h_{k-1} + h_{k-2}}\right)\left(\frac{h_k}{h_k + h_{k-1} + h_{k-2}}\right)\left(\frac{h_k}{h_{k-1}} + \frac{h_k}{h_{k-2}}\right)\right] \\
\tilde{\alpha}_3 &= -\frac{1}{\tilde{\alpha}}\left(\frac{h_k + h_{k-1}}{h_{k-1} + h_{k-2}}\right)\left(\frac{h_k}{h_k + h_{k-1} + h_{k-2}}\right)\left(\frac{h_k}{h_{k-2}}\right),
\end{aligned}$$

with

$$\tilde{\alpha} = 1 + \frac{h_k}{h_k + h_{k-1}} + \frac{h_k}{h_k + h_{k-1} + h_{k-2}}. \tag{4.12}$$

Here, h_k, h_{k-1}, h_{k-2} are the step sizes.

4.2.3 Error estimator

General purpose multistep codes usually estimate the local truncation error to control the step size and the order of the method. In general this error can be estimated by approximating $x^{(p+1)}$ using divided differences, where p is the order of the method.
For the BDF methods the local truncation error can be written as in [47]:

$$h_k \dot{\omega}_k(t_k)[x_k, x_{k-1}, \ldots, x_{k-p}], \tag{4.13}$$

where

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

and

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

for $\psi_j(k) := t_k - t_{k-j}$.
Having the local truncation error for the BDF methods expressed as (4.13) will allow us to compute it directly for low rank factors approximating the solution of DREs see Section 4.2.7.

4.2.4 Adaptive control

In most applications, varying the step size 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.
Algorithm 4.2.1 is similar to the one which underlies the program DASSL of L.R. Petzold [97], the difference here is that error estimators, which we used to decide whether to accept the current step or to redo this with a smaller step size, will be computed using (4.13) instead of using the predictor polynomials involving the steps $p-1$, p, $p+1$; see [31, Algorithm on p. 373].

In Algorithm 4.2.1 Tol represents the desired integration error and $\rho < 1$ is a safety factor, usually chosen as 0.9.
As is noted in [31] the prediction of the next step size is based on consistency error estimates for equidistant meshes and hence works, in some sense with the fiction that the current step size belongs to an equidistant mesh. Several authors proposed changes to improve the robustness of the method [47, 116]. The robustness of the method presented here can be improved taking into account the past two steps in the framework of the control theoretic interpretation of the step size [31].

4.2.5 Application to large-scale DREs

In this section we will show how to apply the step and order selection strategy described before to large-scale differential Riccati equations of the form (4.1)

Algorithm 4.2.1 Step and order control for BDF methods

Require: We are at time t_j, step size h_j and order p.

1: Compute predictor values $x^\nu(t_{j+1})$, $\nu = p-1, p, p+1$.

2: Compute local error estimates $\epsilon_\nu(t_{j+1})$, $\nu = p-1, p, p+1$, based on (4.13).

3: Compute the predicted step sizes

$$h_{j+1}^{(\nu)} = \sqrt[\nu+1]{\frac{\rho \cdot Tol}{|\epsilon_\nu(t_{j+1})|}} h_j, \quad \nu = p-1, p, p+1.$$

4: If at least one of the error estimators satisfies $|\epsilon_\nu(t_{j+1})| \leq Tol$ then choose the index $\nu \in \{\, p-1,\, p,\, p+1\}$ belonging to the smallest error estimate and set

$$x_{j+1} = x^\nu(t_{j+1}), \qquad p = \nu,$$

and the new step size h_{j+1} is determined by

$$h_{j+1} = h_{j+1}^{(new)} = \max(h_{j+1}^{(p-1)}, h_{j+1}^{(p)}, h_{j+1}^{(p+1)}).$$

5: If none of the error estimates satisfies $|\epsilon_\nu(t_{j+1})| \leq Tol$, repeat the process with the corrected step size and order

$$h_{j+1}^{(p)} < h_j,$$

arising in LQR for semi-discretized partial differential equations. This section is essentially contained in [21].

We briefly describe the BDF method for DREs in matrix-valued form similar to [38]. We will then discuss how this scheme can be implemented for large-scale problems. Let us consider

$$F(t, X(t)) \equiv Q(t) + X(t)A(t) + A^T(t)X(t) - X(t)S(t)X(t), \tag{4.14}$$

where $t \in [t_0, t_f]$ and $Q(t)$, $A(t)$, $S(t) \in \mathbb{R}^{n\times n}$, as before, are piecewise continuous locally bounded matrix-valued functions. The fixed-coefficients BDF methods (4.10) applied to the DRE (4.1) 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 4.1.
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 ARE for X_{k+1},

$$\begin{aligned} (h\beta Q_{k+1} &- \textstyle\sum_{j=0}^{p-1} \alpha_j X_{k-j}) + (h\beta A_{k+1} - \tfrac{1}{2}I)^T X_{k+1} + \\ &+ X_{k+1}(h\beta A_{k+1} - \tfrac{1}{2}I) - X_{k+1}(h\beta S_{k+1})X_{k+1} = 0, \end{aligned} \tag{4.15}$$

that can be solved via any method for AREs. Assuming that

$$\begin{aligned} Q_k &= C_k^T C_k, \qquad C_k \in \mathbb{R}^{p\times n}, \\ S_k &= B_k B_k^T, \qquad B_k \in \mathbb{R}^{n\times m}, \\ X_k &= Z_k Z_k^T, \qquad Z_k \in \mathbb{R}^{n\times z_k}, \end{aligned} \tag{4.16}$$

the ARE (4.15) can be written as

$$\begin{aligned} \hat{C}_{k+1}^T\hat{C}_{k+1} &+ \hat{A}_{k+1}^T Z_{k+1}Z_{k+1}^T + Z_{k+1}Z_{k+1}^T\hat{A}_{k+1} \\ &- Z_{k+1}Z_{k+1}^T\hat{B}_{k+1}\hat{B}_{k+1}^T Z_{k+1}Z_{k+1}^T = 0, \end{aligned} \tag{4.17}$$

where

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

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 [6, 93], i.e., it can be well approximated by a *low rank factor* (LRF) Z_k with $z_k \ll n$ for all times. If $z_k \ll n$ for all times, and (4.17) 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. 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 (4.17) 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. We will show how this can be exploited in the ARE solver below.

In our numerical implementation, we benefit from recent algorithmic progress in solving large-scale AREs resulting from semi-discretized control problems for AREs [15, 16, 19]. We will discuss the details of this approach in the next section, which is essentially contained in [21].

4.2.6 Numerical solution of AREs

Since the ARE (4.15) is a nonlinear system of equations, it is quite natural to apply Newton's method to find its solutions. This approach has been investigated; details and further references can be found in [14, 74, 86, 96, 105]. To make the derivation more concise, we will use in this section the generic form of an ARE as it arises in LQR and LQG problems, given by

$$0 \;=\; \mathcal{F}(P) \;:=\; C^T C + A^T P + PA - PBB^T P. \tag{4.18}$$

The case important here, i.e., constant terms of the form $\hat{C}\hat{C}^T - \tilde{C}^T\tilde{C}$, will be explained in Remark 4.2.2 below.

Observing that the (Frechét) derivative of $\mathcal{F}$ at P is given by the Lyapunov operator

$$\mathcal{F}_P^{'} : Q \to (A - BB^T P)^T Q + Q(A - BB^T P),$$

Newton's method for AREs can be written as

$$\begin{aligned} N_\ell &:= -\left(\mathcal{F}_{P_\ell}^{'}\right)^{-1} \mathcal{F}(P_\ell), \\ X_{\ell+1} &:= X_\ell + N_\ell. \end{aligned}$$

Then one step of the Newton iteration for a given starting matrix P_0 can be implemented as in Algorithm 4.2.2.

We assume exits P_0 such that $A - BB^T P_0$ is stable. (In the applications considered here, we can use the fact that for a small time step, the approximate solution $X_k \approx X(t_k)$ will in general be a good stabilizing starting value.)

Algorithm 4.2.2 One step of Newton's method for AREs

Require: P_ℓ, such that A_ℓ is stable.

1: $A_\ell \leftarrow A - BB^T P_\ell$
2: Solve the Lyapunov equation $A_\ell^T N_\ell + N_\ell A_\ell = -\mathcal{F}(P_\ell)$.
3: $P_{\ell+1} \leftarrow P_\ell + N_\ell$

Then all A_ℓ are stable and the iterates P_ℓ converge to P_* quadratically. (Here: $P_* = X_{k+1} \approx X(t_{k+1})$.) In order to make this iteration work for large-scale problems, we need a Lyapunov equation solver that employs the structure of A_ℓ as "sparse + low rank perturbation" by avoiding to form A_ℓ explicitly, and which computes a low rank approximation to the solution of the Lyapunov equation.

For the problems under consideration the spectrum of the positive semidefinite matrix $P_\ell = Z_\ell Z_\ell^T$ often decays to zero rapidly. A typical situation is given in Figure 4.1, where the eigenvalues of P_ℓ for an LQR problem arising from a finite-element discretization of a one-dimensional heat control problem are plotted.

For an eigenvalue decay as in Figure 4.1, we expect that P_ℓ can be approximated accurately by a factorization ZZ^T for some $Z \in \mathbb{R}^{n\times r}$ with $r \ll n$. Such an approximation is obtained by truncating the spectral decomposition $P_\ell = \sum_{j=1}^n \lambda_j z_j z_j^T$ after the first r terms. Here, the eigenvalues λ_j are ordered by decreasing magnitude and z_j is an eigenvector of P_ℓ corresponding to λ_j.

There are partial results explaining the decay of the eigenvalues of Lyapunov and Riccati solutions; bounds and estimates for the decay are given in [6, 93]. Structural information of the underlying physical problem has not yet been incorporated into the analysis. Such information might shed more light on the existence of accurate low rank approximations.

A relevant method, based on this observation, is derived in detail in [19, 91] and is described in the following.

First, we re-write Newton's method for AREs such that the next iterate is computed directly from the Lyapunov equation in Step 2,

$$A_\ell^T P_{\ell+1} + P_{\ell+1} A_\ell = -C^T C - P_\ell B B^T P_\ell =: -W_\ell W_\ell^T. \tag{4.19}$$

Assuming $P_\ell = Z_\ell Z_\ell^T$ for $\operatorname{rank}(Z_\ell) \ll n$ and observing that $\operatorname{rank}(W_\ell) \leq m + p \ll n$, we need only a numerical method to solve Lyapunov equations having a low rank right hand side which returns a low rank approximation to the (Cholesky) factor of its solution. For this purpose, we can use a modified version of the *alternating directions implicit (ADI)* method for Lyapunov equations of the form

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

with F stable, and $W \in \mathbb{R}^{n\times n_w}$, then the ADI iteration can be written as [114]

$$\begin{aligned} (F^T + p_j I) Y_{(j-1)/2} &= -WW^T - Y_{j-1}(F - p_j I), \\ (F^T + \overline{p_j} I) Y_j^T &= -WW^T - Y_{(j-1)/2}(F - \overline{p_j} I), \end{aligned} \tag{4.20}$$

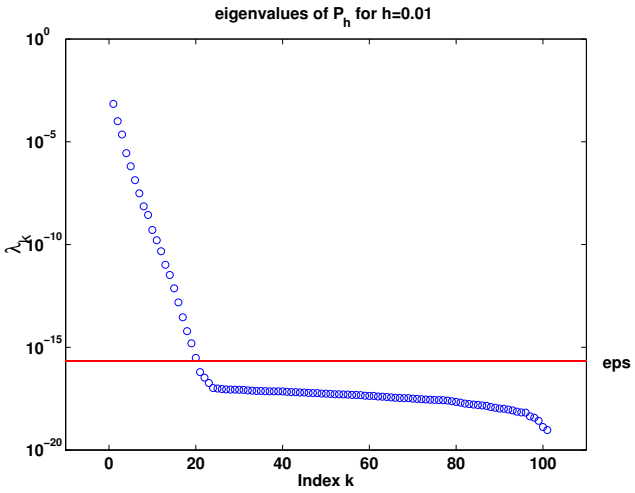


Figure 4.1: Decay of eigenvalues of P_h in the stabilizing Riccati solution. The eigenvalues below the *eps*-line can be set to zero without introducing any significant error in the spectral decomposition of P_h. With increased dimension, the number of eigenvalues larger than machine precision (almost) does not increase.

where $\overline{p}$ denotes the complex conjugate of $p \in \mathbb{C}$. If the shift parameters p_j are chosen appropriately, then $\lim_{j\to\infty} Y_j = Y$ with a superlinear convergence rate. 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 [19, 80, 91] for several variants of this method). The method is described in Algorithm 4.2.3.

Algorithm 4.2.3 LRCF 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 [29, 56] can be applied to reduce the number of columns in $Z_{i_{\max}}$ without adding significant error.

Remark 4.2.1 *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, [29] we can apply a colum compression technique without adding significant error. 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.*

For an implementation of this method, we need a strategy to select the shift parameters p_j. We discuss this problem in detail in Section 4.4.
Since A_ℓ is stable for all ℓ we can apply the modified ADI iteration to (4.19). Then, $W_\ell = \begin{bmatrix} C^T & P_\ell B \end{bmatrix}$ and hence, $n_w = m + p$, so that usually $n_w \ll n$.

Remark 4.2.2 *The solution of the AREs (4.17) arising for BDF methods with $p > 1$, where the constant term is replaced by*

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

as described in the last section, one can to split the Lyapunov equation (4.19) into the two equations

$$\begin{aligned} A_\ell^T \hat{P}_{\ell+1} + \hat{P}_{\ell+1} A_\ell &= -\hat{C}^T \hat{C} - P_\ell B B^T P_\ell, \\ A_\ell^T \tilde{P}_{\ell+1} + \tilde{P}_{\ell+1} A_\ell &= -\tilde{C}^T \tilde{C}. \end{aligned}$$

Then $P_{\ell+1} = \hat{P}_{\ell+1} - \tilde{P}_{\ell+1}$. The two Lyapunov equations can be solved simultaneously by the factored ADI iteration as the linear systems of equations to be solved in each step have the same coefficient matrices.

Note that Algorithm 4.2.3 can be implemented in real arithmetic by combining two steps, even if complex shifts need to be used, which may be the case if A_ℓ is nonsymmetric. A complexity analysis of the factored ADI method depends on the method used for solving the linear systems in each iteration step. If applied to $F = A_\ell^T$ from (4.19), we have to deal with the situation that A_ℓ is a shifted sparse matrix plus a low rank perturbation. If we can solve the shifted linear system of equations in (4.20) efficiently, the low rank perturbation can be dealt with using the Sherman-Morrison-Woodbury formula [54] in the following way: let ℓ be the index of the Newton iterates and let j be the index of the ADI iterates used to solve the ℓth Lyapunov equation, respectively, and set $K_\ell := B^T P_\ell$. Then

$$\begin{aligned} \left(F^T + p_j^{(\ell)} I_n\right)^{-1} &= \left(A + p_j^{(\ell)} I_n - B K_\ell\right)^{-1} \\ &= \left(I_n + L_\ell B (I_m - K_\ell L_\ell B)^{-1} K_\ell\right) L_\ell, \end{aligned}$$

where $L_\ell := (A + p_j^{(\ell)} I_n)^{-1}$. Hence, all linear systems of equations to be solved in one iteration step have the same coefficient matrix $A+p_j^{(\ell)} I_n$. If $A+p_j^{(\ell)} I_n$ is a banded matrix or can be re-ordered to become banded, then a direct solver can be employed. If workspace permits, it is desirable to compute a factorization of $A + p_j^{(\ell)} I_n$ for each different shift parameter beforehand (usually, very few parameters are used). These factorizations can then be used in each iteration step of the ADI iteration. In particular, if A is symmetric positive definite, as will be the case in many applications from PDE constrained optimal control problems, and can be re-ordered in a narrow band matrix, then each factorization requires $\mathcal{O}(n)$ flops, and the total cost $\mathcal{O}(\ell_{\max} \max(j_{\max})n)$ scales with n as desired. If iterative solvers are employed for the linear systems, it should be noted that only one Krylov space needs to be computed (see [80] for details) and hence we obtain an efficient variant of the factored ADI iteration.

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 $\|FZ_jZ_j^T + Z_jZ_j^TF^T + WW^T\|$; see [93] for an efficient way to compute the Frobenius norms of the residuals. On the other hand, the Newton iteration inside Algorithm 4.2.4 (steps 7–12), is usually stopped when

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

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 [9]. 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 the accuracy provided by the BDF method.
The standard implementation of the BDF methods for DREs is sketched in Algorithm 4.2.4.
In the next section we will apply step size and order control for the DRE in terms of the low rank factors (LRF) of the approximated solution.

4.2.7 Step size and order control

If we want to vary the step and order of a LMM method, the solution values at past times on an equidistant mesh are needed. Using the variable-coefficient BDF methods (4.11) we avoid to compute these values. Note that this method applied to (4.1) yields an equation similar to (4.17) 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 because we are working on large-scale problems. On the other hand, for fixed-coefficient BDF methods we can approximate these values using an interpolating polynomial described by Neville's algorithm, which in matrix value form can be expressed as:
Assuming that

Algorithm 4.2.4 LRF BDF method of order p

Require: $A(t)$, $S(t)$, $Q(t)$, $\in \mathbb{R}^{n\times n}$ smooth matrix-valued functions satisfying (4.16), $t \in [a,b]$, and h step size.
Ensure: (Z_i, t_i) such that $X_i \approx Z_i Z_i^T$.

$t_0 = a$.
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 4.2.3 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

Algorithm 4.2.5 Neville's Algorithm

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

$T_{i,o} := X_i \qquad 0 \le i \le n.$
$T_{i,k} := \frac{(t-t_{i-k})T_{i,k-1} - (t-t_i)T_{i-1,k-1}}{t_i - t_{i-k}} \qquad 0 \le i < k \le n.$

Algorithm 4.2.6 LRF Neville's Algorithm

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

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

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

we get

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

so that

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

Hence Algorithm 4.2.5 can be written in terms of the LRFs as in Section 4.2.5, see Algorithm 4.2.6.
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 4.2.6 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 that the solution values at past times are approximated we are ready to apply Algorithm 4.2.1. In step one we need to compute local error estimators, this can be done using (4.13) and computing the divided differences directly for the factors, see Algorithm 4.2.7.

Analogous to Algorithm 4.2.6, Algorithm 4.2.7 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 (4.13).

Algorithm 4.2.7 LRF Divided differences

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

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

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

4.3 Rosenbrock methods

4.3.1 Introduction

Linear multistep methods require fewer function evaluation per step than one step methods, and they allow a simpler, more 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 see,e.g., [31, 58]. We focus here on the Rosenbrock methods, which are DIRK type methods. 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 literature, variants of the Rosenrock method are discussed in which the Jacobian matrix is retained over several steps or even replaced by an approximation which renders the linear system cheaper. Methods constructed in this way were first studied by T. Steihaug and A. Wolfbrand in 1979. Since they denoted the inexact Jacobi matrix by "W", these methods are often called W-methods.
Rosenbrock methods are very attractive for several reasons, among the most popular ones we cite:

- Like implicit methods, Rosenbrock methods require the solution of a linear system of equations; however, unlike implicit methods, they do not require the added burden of iteration to accomplish the task of solving the system and therefore they are more easy to implement.
- They are suitable for parallelization [33, 113].
- They possess excellent stability properties, as they can be made A-stable or L-stable.
- They are computationally efficient while preserving positivity of the solutions.
- They are of one-step type which allows a rapid change of step size.
- They are also applicable to implicit systems of the form $M\dot{y} = f(y)$.

These methods have already proven to be very effective in some applications like chemical kinetics [30, 50, 113], and several variants of these methods have been proposed, e.g., in [50] the coefficients of the Rosenbrock method are chosen common to an explicit Runge-Kutta method of order 4. The result is an embedded Rosenbrock integrator of order 4, i.e., a Rosenbrock integrator that contains an explicit Runge-Kutta method embedded, that switches from one to the other solver when the solution leaves a stiff domain and enters a nonstiff domain or vice versa.

4.3.2 Rosenbrock schemes

In the following we describe some Rosenbrock schemes which we will apply to DREs in the next section. First of all, let us define a general s-stage Rosenbrock method for an autonomous ODE system:

$$\begin{array}{rcl} k_i & = & hf\big(x_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j\big) + hJ \sum_{j=1}^{i} \gamma_{ij} k_j, \quad i = 1, \ldots, s, \\ x_{n+1} & = & x_n + \sum_{j=1}^{s} b_j k_j, \end{array} \tag{4.21}$$

where α_{ij}, γ_{ij}, b_j are the determining coefficients, J is the Jacobian matrix $f'(x_n)$, and h is the step size. Each stage of this method consists of a system of linear equations with unknowns k_i. Of special interest are methods for which $\gamma_{11} = \cdots = \gamma_{ss} = \gamma$, so that only one LU-decomposition per step is needed. Note that for $J = 0$ an explicit Runge-Kutta method is recovered.
The non-autonomous ODE system

$$\dot{x} = f(t, x) \tag{4.22}$$

can be converted to autonomous form by adding $\dot{x} = 1$. If the method (4.21) is applied to the augmented system, the components corresponding to the x-variable can be computed explicitly and we arrive at

$$\begin{array}{rcl} k_i & = & hf\big(t_n + \alpha_i h, x_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j\big) + \gamma_i h^2 \frac{\partial f}{\partial t}(t_n, x_n) \\ & & + h \frac{\partial f}{\partial x}(t_n, x_n) \sum_{j=1}^{i} \gamma_{ij} k_j, \\ x_{n+1} & = & x_n + \sum_{j=1}^{s} b_j k_j, \end{array} \tag{4.23}$$

where the additional coefficients are given by

$$\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \qquad \gamma_i = \sum_{j=1}^{i} \gamma_{ij}.$$

This definition was taken from [58], refer to it and references therein for a detailed explanation.
Note that for an autonomous system and $s = 1$, the linearly implicit Euler method is recovered:

$$\begin{array}{rcl} x_{n+1} & = & x_n + hk_1, \\ (I - hJ)k_1 & = & f(x_n), \end{array} \tag{4.24}$$

where J is the Jacobian and the coefficients are chosen as

$$b_1 = 1, \qquad \gamma = 1, \qquad \alpha_{11} = 0.$$

The method is of order $p = 1$ and the stability function is the same as the for implicit Euler method.
In [30] a second order method (for autonomous ODE systems) is described for application to atmospheric dispersion problems describing photochemistry, advective, and turbulent diffusive transport. The scheme is written in the form

$$\begin{aligned} x_{n+1} &= x_n + \frac{3}{2}hk_1 + \frac{1}{2}hk_2, \\ (I - \gamma hJ)k_1 &= f(x_n), \\ (I - \gamma hJ)k_2 &= f(x_n + hk_1) - 2k_1, \end{aligned} \tag{4.25}$$

where J is the Jacobian matrix $f'(x_n)$ or an approximation thereof. The parameter γ appears in the stability function of the method,

$$R(z) = \frac{1 + (1 - 2\gamma)z + (\frac{1}{2} - 2\gamma + \gamma^2)z^2}{(1 - \gamma z)^2}.$$

If $\gamma \geq 1/4$, the method is A-stable. L-stability is achieved using $\gamma = 1 + 1/\sqrt{2}$. It is pointed out by the authors that the method is capable of integrating with large a priori described step sizes.

Non-autonomous systems. According to (4.23) the scheme (4.25), for the non-autonomous case, can be written as

$$\begin{aligned} x_{n+1} &= x_n + \frac{3}{2}hk_1 + \frac{1}{2}hk_2, \\ (I - \gamma hJ)k_1 &= f(t_n, x_n) + \gamma hf_t, \\ (I - \gamma hJ)k_2 &= f(t_n + h, x_n + hk_1) - 2k_1 - \gamma hf_t, \end{aligned} \tag{4.26}$$

where

$$J = \frac{\partial f}{\partial x}(t_n, x_n), \qquad f_t = \frac{\partial f}{\partial t}(t_n, x_n). \tag{4.27}$$

The linearly implicit Euler method (4.24) for non-autonomous systems becomes

$$\begin{aligned} x_{n+1} &= x_n + hk_1, \\ (I - hJ)k_1 &= f(x_n) + hf_t \end{aligned} \tag{4.28}$$

with J and f_t as in (4.27).

4.3.3 Application to DREs

As before we consider symmetric DREs of the form (4.1) and $F(t, X(t))$ as in (4.14). Let us denote $F(t, X(t)) \equiv F(X(t)) := \mathcal{F}(X)$.

The Jacobian $J = \mathcal{F}'(X_k)$ in (4.26) is given by the (Frechét) derivative of $\mathcal{F}$ at X_k represented by the Lyapunov operator

$$\mathcal{F}'(X_k) : U \to (A_k - S_k X_k)^T U + U(A_k - S_k X_k),$$

where $X_k \approx X(t_k)$, $A_k = A(t_k)$, $S_k = S(t_k)$ and $U \in \mathbb{R}^{n\times n}$.
Let us denote $F_{t_k} = \frac{\partial F}{\partial t}(t_k, X(t_k))$, which is given by

$$\begin{aligned} F_{t_k} &= \dot{Q}_k + A_k^T \dot{X}_k + \dot{A}_k^T X + \dot{X}_k A_k + X_k \dot{A}_k - \dot{X}_k S_k X_k \\ &\quad -X_k \dot{S}_k X_k - X_k S_k \dot{X}_k, \end{aligned} \tag{4.29}$$

where $\dot{Q}_k = \frac{dQ}{dt}(t_k)$, $\dot{A}_k = \frac{dA}{dt}(t_k)$, $\dot{X}_k = \frac{dX}{dt}(t_k)$, and $\dot{S}_k = \frac{dS}{dt}(t_k)$. Later we explain how these derivatives can be approximated.
The application of the linear implicit Euler method (4.28), as a matrix-valued algorithm, to the DRE (4.1) yields

$$\begin{aligned} X_{k+1} &= X_k + hK_1, \\ K_1 - h(\mathcal{F}'(X_k))(K_1) &= F(X_k) + hF_{t_k}. \end{aligned} \tag{4.30}$$

We use K_i instead of k_i, $i = 1, 2$ because now they represent $n \times n$ matrices.
Replacing $\mathcal{F}'(X_k)$ in (4.30) we obtain

$$K_1 - h(A_k - S_k X_k)^T K_1 - hK_1(A_k - S_k X_k) = F(X_k) + hF_{t_k},$$

and re-arranging terms yields

$$(h(A_k - S_k X_k) - \frac{1}{2}I)^T K_1 + K_1(h(A_k - S_k X_k) - \frac{1}{2}I) = -F(X_k) - hF_{t_k}. \tag{4.31}$$

Denoting $\bar{A}_k = h(A_k - S_k X_k) - \frac{1}{2}I$, we can write the method as:

$$X_{k+1} = X_k + hK_1, \tag{4.32}$$

$$\bar{A}_k^T K_1 + K_1 \bar{A}_k = -F(X_k) - hF_{t_k}. \tag{4.33}$$

Hence, one Lyapunov equation (4.33) has to be solved in every step.
If we write $F(X_k)$ as

$$\begin{gathered} \big(A_k - S_k X_k - \tfrac{1}{2h}I\big)^T X_k + X_k\big(A_k - S_k X_k - \tfrac{1}{2h}I\big) \\ +Q_k + X_k S_k X_k + \tfrac{1}{h}X_k, \end{gathered}$$

and denoting $\tilde{A}_k = A_k - S_k X_k - \frac{1}{2h}I$, then we can re-write the linear implicit Euler method (4.32)–(4.33) such that the next iterate is computed directly from the Lyapunov equation (4.33),

$$\tilde{A}_k^T X_{k+1} + X_{k+1}\tilde{A}_k = -Q_k - X_k S_k X_k - \frac{1}{h}X_k - hF_{t_k}. \tag{4.34}$$

The application of the Rosenbrock method (4.26), as a matrix-valued algorithm, to the DRE (4.1) yields

$$\begin{aligned} X_{k+1} &= X_k + \frac{3}{2}hK_1 + \frac{1}{2}hK_2, \\ K_1 - \gamma h(\mathcal{F}'(X_k))(K_1) &= F(X_k) + \gamma h F_{t_k}, \qquad (4.35) \\ K_2 - \gamma h(\mathcal{F}'(X_k))(K_2) &= F(t_k + h, X_k + hK_1) - 2K_1 - \gamma h F_{t_k}. \qquad (4.36) \end{aligned}$$

Denoting $\hat{A}_k = \gamma h(A_k - S_k X_k) - \frac{1}{2}I$, $t_{k+1} = t_k + h$ and rewriting (4.35), (4.36) similar to (4.33), we can write the method as:

$$\begin{aligned} X_{k+1} &= X_k + \frac{3}{2}hK_1 + \frac{1}{2}hK_2, \qquad (4.37) \\ \hat{A}_k^T K_1 + K_1 \hat{A}_k &= -F(X_k) - \gamma h F_{t_k}, \qquad (4.38) \\ \hat{A}_k^T K_2 + K_2 \hat{A}_k &= -F(t_{t+1}, X_k + hK_1) + 2K_1 + \gamma h F_{t_k}. \qquad (4.39) \end{aligned}$$

Hence, two Lyapunov equations (4.38), (4.39) have to be solved in every step. Our analysis can be extended to a general s-stage Rosenbrock method which will require the solution of s Lyapunov equations in every step. For the case in which the coefficient matrices of Lyapunov equations are dense, the Bartels-Stewart method [13] can be applied for solving the equations. Note that only one Schur decomposition is needed therefore, the cost is almost that of solving one Lyapunov equation.
Rewriting the right hand side of (4.39) as

$$\begin{aligned} &-F(t_{k+1}, X_k) + \gamma h F_{t_k} - h^2 K_1 S_{k+1} K_1 \\ &-(h(A_{k+1} - S_{k+1}X_k) - I)^T K_1 - K_1(h(A_{k+1} - S_{k+1}X_k) - I), \end{aligned} \qquad (4.40)$$

and noting that it is more efficient to solve an additional Lyapunov equation (with the same coefficient matrix $\hat{A}_k$) in which the right hand side is chosen as the common factor of the right hand sides of (4.38)–(4.39) and afterwards recover the original solution, than solve (4.38)–(4.39) separately. The standard implementation of the method (4.37)–(4.39) can then be sketched as in Algorithm 4.3.1.

Remark 4.3.1 *We point out that the intermediate approximation $X_k + hK_1$ corresponds to the application of the linearly implicit Euler method at t_{k+1}. This first order approximation can be used to estimate the local error for step size control as outlined in Algorithm 4.3.2. We follow [31, Alg. 5.2, p. 194], as is explained there a bound on the step increase has to be introduced. Hence, the growth of the step is limited by a factor $q > 1$ or by the maximum step size allowed $h_{\max}$.*

Autonomous DRE. Note that for autonomous DREs, i.e., DREs in which matrices $Q(t)$, $A(t)$, $R(t)$ are constant, the second order Rosenbrock method

Algorithm 4.3.1 Rosenbrock method of order two

Require: $Q(t)$, $A(t)$, $S(t)$, $\in \mathbb{R}^{n\times n}$ are piecewise continuous locally bounded matrix-valued functions $t \in [a,b]$, X_0, and h step size.

Ensure: (X_i, t_i) such that $X_i \approx X(t_i)$.

$t_0 = a$.
for $k = 0$ to $\lceil \frac{b-a}{h} \rceil$ **do**
  $t_{k+1} = t_k + h$.
  $\hat{A}_k = \gamma h(A - RX_k) - \frac{1}{2}I$.
  Solve Lyapunov equation $\hat{A}_k^T K_{11} + K_{11}\hat{A}_k = -F(X_k)$.
  Solve Lyapunov equation $\hat{A}_k^T K_{12} + K_{12}\hat{A}_k = -F_{t_k}$.
  $K_1 = K_{11} + \gamma h K_{12}$.
  Solve Lyapunov equation

$$\begin{aligned} \hat{A}_k^T K_{21} \quad &+ \quad K_{21}\hat{A}_k = -(h(A_{k+1} - S_{k+1}X_k) - I)^T K_1 \\ &- \quad K_1(h(A_{k+1} - S_{k+1}X_k) - I) - h^2 K_1 S_{k+1} K_1 - F(t_{k+1}, X_k). \end{aligned}$$

  $K_2 = K_{21} - \gamma h K_{12}$.
  $X_{k+1} = X_k + \frac{3}{2}hK_1 + \frac{1}{2}hK_2$.
end for

Algorithm 4.3.2 Step size control for Rosenbrock method of order two

Require: Let h_0 be the initial step size, $[a,b]$ the integration interval, X_0 the initial condition, $\rho < 1$ and $q > 1$ safety parameters, Tol desired integration error, and $h_{\max}$ maximum step size allowed.

$k = 0$.
$t_0 = a$.
while $t_k < b$ **do**
  $t = t_k + h_k$.
  Compute K_1 by (4.38).
  $\hat{Y}_{k+1} = X_k + hK_1$.
  Compute K_2 by (4.39).
  $Y_{k+1} = \frac{3}{2}hK_1 + \frac{1}{2}hK_2$.
  $\epsilon_k = \left\| Y_{k+1} - \hat{Y}_{k+1} \right\|$.
  $h = \min(qh_k, h_{\max}, \sqrt[3]{\frac{\rho \cdot Tol}{\epsilon_k}} h_k)$.
  if $\epsilon_k < Tol$ **then**
    $t_{k+1} = t$.
    $X_{k+1} = Y_{k+1}$.
    $h_{k+1} = \min(h, b - t_{k+1})$.
    $k = k + 1$.
  else
    $h_k = h$.
  end if
end while

can be written as

$$\begin{aligned} X_{k+1} &= X_k + \frac{3}{2}hK_1 + \frac{1}{2}hK_2, && (4.41)\\ \hat{A}_k^T K_1 + K_1 \hat{A}_k &= -F(X_k), && (4.42)\\ \hat{A}_k^T K_{21} + K_{21} \hat{A}_k &= -(h(A - SX_k) - I)^T K_1 \\ &\quad -K_1(h(A - SX_k) - I) - h^2 K_1 S K_1, && (4.43)\\ K_2 &= K_1 + K_{21}. && (4.44) \end{aligned}$$

where $\hat{A}_k = \gamma h(A - SX_k) - \frac{1}{2}I$. If in addition we chose $\gamma = 1$ (A-stability is achieved for $\gamma \geq \frac{1}{4}$), then the method results in

$$\begin{aligned} X_{k+1} &= X_k + \frac{3}{2}hK_1 + \frac{1}{2}hK_2, && (4.45)\\ \hat{A}_k^T K_1 + K_1 \hat{A}_k &= -F(X_k), && (4.46)\\ \hat{A}_k^T K_2 + K_2 \hat{A}_k &= -h^2 K_1 S K_1 - K_1. && (4.47) \end{aligned}$$

The linearly implicit Euler method becomes

$$\tilde{A}_k^T X_{k+1} + X_{k+1} \tilde{A}_k = -Q - X_k S X_k - \frac{1}{h} X_k, \tag{4.48}$$

where $\tilde{A}_k = A - SX_k - \frac{1}{2h}I$.

4.3.4 Low rank Rosenbrock method

We focus on solving DREs arising in optimal control for parabolic partial differential equations. Typically the coefficient matrices of the DRE arising from these control problems have a certain structure (e.g. sparse, symmetric or low rank). Thus, the solution of the resulting Lyapunov equation with the Bartels-Stewart method is not feasible. In this section we show that it is possible to efficiently implement Rosenbrock methods for large-scale DREs based on a low rank version of the alternating direction implicit (ADI) iteration for Lyapunov equations [19, 80, 92].

Linearly implicit Euler method. Let us first consider the linearly implicit Euler method for autonomous DREs (4.48) and assume,

$$\begin{aligned} Q &= C^T C, \qquad C \in \mathbb{R}^{p \times n}, \\ S &= BB^T, \qquad B \in \mathbb{R}^{n \times m}, \\ X_k &= Z_k Z_k^T, \qquad Z_k \in \mathbb{R}^{n \times z_k}. \end{aligned} \tag{4.49}$$

with p, m, $z_k \ll n$. If we denote $N_k = [\, C^T \; Z_k(Z_k^T B) \; \sqrt{h^{-1}} Z_k \,]$, then the Lyapunov equation (4.48) results in

$$\tilde{A}_k^T X_{k+1} + X_{k+1} \tilde{A}_k = -N_k N_k^T, \tag{4.50}$$

where $\tilde{A}_k = A - B(Z_k(Z_k^T B))^T - \frac{1}{2h}I$. 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 (Algorithm 4.2.3, in Section 4.2.6) to solve (4.50).
The application of Algorithm 4.2.3 to (4.50) will ensure low rank factors Z_{k+1}, of X_{k+1}, such that $X_{k+1} = Z_{k+1}Z_{k+1}^T$, where $Z_{k+1} \in \mathbb{R}^{n \times z_{k+1}}$ with $z_{k+1} \ll n$. This is described in Algorithm 4.3.3.

Algorithm 4.3.3 LRF linearly implicit Euler method

Require: $A \in \mathbb{R}^{n\times n}$, B, C, Z_0 satisfying (4.49), $t \in [a, b]$, and h step size.
Ensure: (Z_i, t_i) such that $X_i \approx Z_i Z_i^T$, $Z_i \in \mathbb{R}^{n\times z_i}$ with $z_i \ll n$.

1: $t_0 = a$.
2: **for** $k = 0$ to $\lceil \frac{b-a}{h} \rceil$ **do**
3: $\quad \tilde{A}_k = A - B(Z_k(Z_k^T B))^T - \frac{1}{2h}I$.
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 $\tilde{A}_k$.
6: $\quad$ Compute Z_{k+1} by Algorithm 4.2.3 such that the low rank factor product $Z_{k+1}Z_{k+1}^T$ approximates the solution of $\tilde{A}_k^T X_{k+1} + X_{k+1}\tilde{A}_k = -N_k N_k^T$.
7: $\quad t_{k+1} = t_k + h$.
8: **end for**

Rosenbrock method of second order. Let us now turn our attention to the method (4.41)-(4.44). As for the linearly implicit Euler method we want to apply the ADI iteration to solve the Lyapunov equations (4.42) and (4.43).
First of all, note that K_1 and K_{21} are computed in every step, so we denote by $K_1 := K_1(k)$ and $K_{21} := K_{21}(k)$ the solution, at step k, of (4.42) and (4.43) respectively.
Hence, (4.41) can be written as

$$X_k = X_0 + h\left(2\sum_{j=0}^{k-1} K_1(j) + \frac{1}{2}\sum_{j=0}^{k-1} K_{21}(j)\right).$$

Moreover, for every k the following equation holds

$$\begin{aligned} 2\textstyle\sum_{j=0}^{k-1} K_1(j) + \frac{1}{2}\sum_{j=0}^{k-1} K_{21}(j) &= \hat{K}_k - \tilde{K}_k \\ &= \hat{T}_k\hat{T}_k^T - \tilde{T}_k\tilde{T}_k^T. \end{aligned} \tag{4.51}$$

In fact, for $k = 0$ let us assume (4.49) and note that,

$$\begin{aligned} A^T Z_0 Z_0^T + Z_0 Z_0^T A &= A^T Z_0(Z_0^T A + Z_0^T) + Z_0(Z_0^T A + Z_0^T) \\ &\quad -A^T Z_0 Z_0^T A - Z_0 Z_0^T, \\ &= (A^T Z_0 + Z_0)(Z_0^T A + Z_0^T) - A^T Z_0 Z_0^T A \\ &\quad -Z_0 Z_0^T, \\ &= (A^T Z_0 + Z_0)(A^T Z_0 + Z_0)^T \\ &\quad -[\, A^T Z_0 \;\; Z_0 \,][\, A^T Z_0 \;\; Z_0 \,]^T. \end{aligned}$$

Denoting $L_0 := A^T Z_0 + Z_0$, $M_0 := [\, A^T Z_0 \; Z_0 \,]$, and $W_0 := Z_0(Z_0^T B)$, then the right hand side of (4.42), at $k = 0$, can be written as

$$- C^T C - L_0 L_0^T + M_0 M_0^T + W_0 W_0^T. \tag{4.52}$$

Then, if we denote $N_0 := [\, C^T \; L_0 \,]$, $U_0 := [\, M_0 \; W_0 \,]$, we can split the Lyapunov equation (4.42), at $k = 0$, into

$$\begin{aligned} \hat{A}_0^T \tilde{K}_1 + \tilde{K}_1 \hat{A}_0 &= -U_0 U_0^T, & (4.53)\\ \hat{A}_0^T \hat{K}_1 + \hat{K}_1 \hat{A}_0 &= -N_0 N_0^T, & (4.54) \end{aligned}$$

where $K_1(0) := \hat{K}_1 - \tilde{K}_1$.
Hence, assuming $\operatorname{rank}(Z_0) \leq z_0 \ll n$ and observing then $\operatorname{rank}(N_0) \leq p + z_0 \ll n$, and $\operatorname{rank}(U_0) \leq 2z_0 + m \ll n$, we can use the modified version of the ADI iteration (Algorithm 4.2.3, in Section 4.2.6) to solve (4.53) and (4.54).
The application of Algorithm 4.2.3 to (4.53) and (4.54) will ensure low rank factors $\tilde{T}_1^0$, and $\hat{T}_1^0$ of $\tilde{K}_1$ and $\hat{K}_1$ respectively, such that $K_1(0) = \hat{T}_1^0 \hat{T}_1^{0T} - \tilde{T}_1^0 \tilde{T}_1^{0T}$ where $\hat{T}_1^0 \in \mathbb{R}^{n \times \hat{t}_0}$, $\tilde{T}_1^0 \in \mathbb{R}^{n \times \tilde{t}_0}$ with $\hat{t}_0$, $\tilde{t}_0 \ll n$. We do not compute explicitly $K_1(0)$, or a low rank factor of it because it will be complex as we want to keep the computation in real arithmetics. Instead, we use the split representation of $K_1(0)$ in the right hand side of (4.43), which can be expressed as

$$\begin{aligned} -h(A^T K_1(0) + K_1(0)A) + h^2 K_1(0) S K_1(0) - 2K_1(0) \\ + h(K_1(0) S X_0 + X_0 S K_1(0)). \end{aligned}$$

Notice that, denoting $\hat{K}_1(0) = \hat{T}_1^0 (\hat{T}_1^0)^T$, then

$$\begin{aligned} X_0 S \hat{K}_1(0) + \hat{K}_1(0) S X_0 &= Z_0 Z_0^T B B^T (Z_0 Z_0^T + \hat{T}_1^0 (\hat{T}_1^0)^T) \\ &\quad + \hat{T}_1^0 (\hat{T}_1^0)^T B B^T (Z_0 Z_0^T + \hat{T}_1^0 (\hat{T}_1^0)^T) \\ &\quad - Z_0 Z_0^T B B^T Z_0 Z_0^T - \hat{T}_1^0 (\hat{T}_1^0)^T B B^T \hat{T}_1^0 (\hat{T}_1^0)^T, \\ &= (Z_0 Z_0^T + \hat{T}_1^0 (\hat{T}_1^0)^T) B B^T (Z_0 Z_0^T + \hat{T}_1^0 (\hat{T}_1^0)^T) \\ &\quad - Z_0 Z_0^T B B^T Z_0 Z_0^T - \hat{T}_1^0 (\hat{T}_1^0)^T B B^T \hat{T}_1^0 (\hat{T}_1^0)^T, \end{aligned}$$

thus $X_0 S \hat{K}_1(0) + \hat{K}_1(0) S X_0$ can be expressed as

$$\begin{aligned} [Z_0(Z_0^T B) + \hat{T}_1^0((\hat{T}_1^0)^T B)][Z_0(Z_0^T B) + \hat{T}_1^0((\hat{T}_1^0)^T B)]^T \\ - [Z_0(Z_0^T B)][Z_0(Z_0^T B)]^T - [\hat{T}_1^0((\hat{T}_1^0)^T B)][\hat{T}_1^0((\hat{T}_1^0)^T B)]^T. \end{aligned} \tag{4.55}$$

Therefore, denoting

$$\begin{aligned} &\hat{L}_0^1 := A^T \hat{T}_1^0 + \hat{T}_1^0, && \tilde{L}_0^1 := A^T \tilde{T}_1^0 + \tilde{T}_1^0, \\ &\hat{M}_0^1 := [\, A^T \hat{T}_1^0 \; \hat{T}_1^0 \,], && \tilde{M}_0^1 := [\, A^T \tilde{T}_1^0 \; \tilde{T}_1^0 \,], \\ &\hat{W}_0^1 := \hat{T}_1^0((\hat{T}_1^0)^T B), && \tilde{W}_0 := \tilde{T}_1^0((\tilde{T}_1^0)^T B), \\ &W_0 := Z_0(Z_0^T B), \end{aligned} \tag{4.56}$$

and re-arranging terms we get

$$U_0^2(U_0^2)^T - N_0^2(N_0^2)^T,$$

where

$$\begin{array}{rcl} U_0^2 & := & [\,\sqrt{h}\tilde{L}_0^1\ \sqrt{h}\hat{M}_0^1\ \sqrt{2}\tilde{T}_1^0\ \sqrt{2h^2+h}\tilde{W}_0^1\ \sqrt{2}h\hat{W}_0^1\ \sqrt{h}(W_0+\hat{W}_0^1)\,], \\ N_0^2 & := & [\,\sqrt{h}\hat{L}_0^1\ \sqrt{h}\tilde{M}_0^1\ \sqrt{2}\hat{T}_1^0\ h(\hat{W}_0^1+\tilde{W}_0^1)\ \sqrt{h}\hat{W}_0^1\ \sqrt{h}(W_0+\tilde{W}_0^1)\,]. \end{array}$$

Assuming that $\mathrm{rank}(\hat{T}_1^0) \leq \hat{r}_0$ and $\mathrm{rank}(\tilde{T}_1^0) \leq \tilde{r}_0$, we observe that $\mathrm{rank}(U_0^2)$, $\mathrm{rank}(N_0^2) \leq 2\tilde{r}_0 + 2\hat{r}_0 + 3m$, which in general we expect to be much smaller than n. In case $\mathrm{rank}(U_0^2)$ ($\mathrm{rank}(N_0^2)$) becomes large a column compression technique can be applied to reduce the number of columns of U_k^2 (N_k^2) without adding a significant error, see Remark 4.2.1. Therefore, we can apply the ADI iteration to solve the Lyapunov equations resulting from splitting (4.43). Let us denote by $\tilde{T}_2^0$ and $\hat{T}_2^0$ the low rank factors given by the ADI iteration. Then, setting

$$\hat{T}_k := \left[\,\sqrt{2}\hat{T}_1^0\ \sqrt{\frac{1}{2}}\hat{T}_2^0\,\right], \qquad \tilde{T}_k := \left[\,\sqrt{2}\tilde{T}_1^0\ \sqrt{\frac{1}{2}}\tilde{T}_2^0\,\right]$$

(4.51) holds for $k = 0$.
Let us now assume that (4.51) holds for a given k. We prove that it holds for $k+1$. If we re-write the right hand side of (4.42) as

$$\begin{array}{rcl} -F(X_k) & = & -F(X_0 + h(\hat{K}_k - \tilde{K}_k)) \\ & = & -F(X_0) - F(h\hat{K}_k) + F(h\tilde{K}_k) + 2h^2\tilde{K}_k S\tilde{K}_k \\ & & +hX_0S\hat{K}_k - hX_0S\tilde{K}_k + h\hat{K}_kSX_0 - h^2\hat{K}_kS\tilde{K}_k \\ & & -h\tilde{K}_kSX_0 - h^2\tilde{K}_kS\hat{K}_k. \end{array}$$

then, $F(h\hat{K}_k)$ and $F(h\tilde{K}_k)$ can be expressed as a sum of low rank matrix products similar to $F(X_0)$ in (4.52). On the other hand, a low rank representation of $X_0S\tilde{K}_k + \tilde{K}_kSX_0$, $X_0S\hat{K}_k + \hat{K}_kSX_0$ and $\hat{K}_kR\tilde{K}_k + \tilde{K}_kR\hat{K}_k$ can be found similar to (4.55). Denoting

$$\begin{array}{ll} \hat{L}_k := A^T\hat{T}_k + \hat{T}_k, & \tilde{L}_k := A^T\tilde{T}_k + \tilde{T}_k, \\ \hat{M}_k := [\,A^T\hat{T}_k\ \hat{T}_k\,], & \tilde{M}_k := [\,A^T\tilde{T}_k\ \tilde{T}_k\,], \\ \hat{W}_k := \hat{T}_k(\hat{T}_k^TB), & \tilde{W}_k := \tilde{T}_k(\tilde{T}_k^TB), \end{array} \tag{4.57}$$

and re-arranging terms the right hand side of (4.42) can be written as

$$U_kU_k^T - N_kN_k^T \tag{4.58}$$

where

$$\begin{array}{rcl} U_k & := & \left[\,M_0\ W_0\ \sqrt{h}\hat{M}_k\ \sqrt{2}h\hat{W}_k\ \sqrt{h}\tilde{L}_k\ (\sqrt{2h^2+h})\tilde{W}_k\ \sqrt{h}(W_0+\hat{W}_k)\right], \\ N_k & := & \left[\,C^T\ L_0\ \sqrt{h}\hat{L}_k\ \sqrt{h}\tilde{M}_k\ \sqrt{h}\tilde{W}_k\ \sqrt{h}(W_0+\tilde{W}_k)\ \sqrt{h}(\hat{W}_k+\tilde{W}_k)\right]. \end{array}$$

So, for every step k, we can split the Lyapunov equation (4.42) into two Lyapunov equations, similar to (4.53) and (4.54). If we assume that $\mathrm{rank}(\hat{T}_k) \leq$

$\tilde{t}_k \ll n$ and $\text{rank}(\tilde{T}_k) \leq \hat{t}_k \ll n$, then $\text{rank}(U_k) \leq 2z_0 + 2\hat{t}_k + 4m + \tilde{t}_k \ll n$ and $\text{rank}(N_k) \leq p + 2z_0 + 2\tilde{t}_k + 3m + \hat{t}_k \ll n$ (in case $\text{rank}(U_k)$ ($\text{rank}(N_k)$) becomes large, a column compression technique can be applied to reduce the number of columns of U_k (N_k), see Remark 4.2.1). Therefore, we are able now to apply the ADI iteration for the resulting equations after splitting (4.42) in every step. Let us denote by $\tilde{T}_1^k$ and $\hat{T}_1^k$ the low rank factors computed by the ADI iteration, at step k.
The right hand side of (4.43) can be written as

$$\begin{gathered} h(A^T K_1(k-1) - K_1(k-1)A) - h^2 K_1(k-1)SK_1(k-1) + 2K_1(k-1) \\ -h(K_1(k-1)SX_0 - X_0 SK_1(k-1)) - h^2(K_1(k-1)S\hat{K}_k - \hat{K}_k SK_1(k-1)) \\ +h^2(K_1(k-1)S\tilde{K}_k - \tilde{K}_k SK_1(k-1)). \end{gathered}$$

Denoting

$$\begin{array}{ll} \hat{L}_k^1 := A^T\hat{T}_1^k + \hat{T}_1^k, & \tilde{L}_k^1 := A^T\tilde{T}_1^k + \tilde{T}_1^k, \\ \hat{M}_k^1 := [\, A^T\hat{T}_1^k \ \hat{T}_1^k\,], & \tilde{M}_k^1 := [\, A^T\tilde{T}_1^k \ \tilde{T}_1^k\,], \\ \hat{W}_k^1 := \hat{T}_1^k((\hat{T}_1^k)^T B), & \tilde{W}_k^1 := \tilde{T}_1^k((\tilde{T}_1^k)^T B), \end{array} \tag{4.59}$$

and re-arranging this becomes

$$U_k^2(U_k^2)^T - N_k^2(N_k^2)^T,$$

where

$$\begin{aligned} U_k^2 &:= [\,\sqrt{h}\tilde{L}_k^1 \ \sqrt{h}\hat{M}_k^1 \ \sqrt{2}\tilde{T}_1^k \ \sqrt{2h^2+h}\tilde{W}_k^1 \ \sqrt{2}h\hat{W}_k^1 \\ &\qquad \sqrt{h}(W_0 + \hat{W}_k^1) \ h(\tilde{W}_k + \tilde{W}_k^1) \ h(\hat{W}_k + \hat{W}_k^1)\,], \\ N_k^2 &:= [\,\sqrt{h}\hat{L}_k^1 \ \sqrt{h}\tilde{M}_k^1 \ \sqrt{2}\hat{T}_1^k \ h(\hat{W}_k^1 + \tilde{W}_k^1) \ \sqrt{h}\hat{W}_k^1 \\ &\qquad \sqrt{h}(W_0 + \tilde{W}_k^1) \ h(\hat{W}_k + \tilde{W}_k^1) \ h(\tilde{W}_k + \hat{W}_k^1)\,]. \end{aligned}$$

As before, let us assume that $\text{rank}(\hat{T}_1^k) \leq \hat{r}_k \ll n$ and $\text{rank}(\tilde{T}_1^k) \leq \tilde{r}_k \ll n$, then $\text{rank}(U_k^2)$, $\text{rank}(N_k^2) \leq 2\tilde{r}_k + 2\hat{r}_k + 5m \ll n$ (again, in case $\text{rank}(U_k^2)$ ($\text{rank}(N_k^2)$) becomes large a column compression technique can be applied) . Hence, we can apply the ADI iteration to solve the Lyapunov equations resulting from splitting (4.43) in every step . Let us denote by $\hat{T}_2^k$ and $\tilde{T}_2^k$ the low rank factors given by the ADI iteration at step k, then

$$\begin{aligned} Z_{k+1}Z_{k+1}^T &= Z_0 Z_0^T + \frac{3}{2}h\sum_{j=0}^{k}(\hat{T}_1^j(\hat{T}_1^j)^T - \tilde{T}_1^j(\tilde{T}_1^j)^T) \\ &\quad + \frac{1}{2}h\sum_{j=0}^{k}(\hat{T}_2^j(\hat{T}_2^j)^T - \tilde{T}_2^j(\tilde{T}_2^j)^T), \\ &= Z_0 Z_0^T + h(\hat{T}_{k+1}\hat{T}_{k+1}^T - \tilde{T}_{k+1}\tilde{T}_{k+1}^T), \end{aligned}$$

where

$$\begin{aligned} \hat{T}_k &:= \left[\, \sqrt{2}[\hat{T}_1^0 \, \hat{T}_1^1 \, \ldots \, \hat{T}_1^k\,] \ \sqrt{\frac{1}{2}}[\hat{T}_2^0 \, \hat{T}_2^1 \, \ldots \, \hat{T}_2^k] \right], \\ \tilde{T}_k &:= \left[\, \sqrt{2}[\tilde{T}_1^0 \, \tilde{T}_1^1 \, \ldots \, \tilde{T}_1^k\,] \ \sqrt{\frac{1}{2}}[\tilde{T}_2^0 \, \tilde{T}_2^1 \, \ldots \, \tilde{T}_2^k] \right], \end{aligned}$$

i.e., (4.51) holds for $k+1$ and therefore the whole iteration can be performed in terms of the low rank factors. The method is sketched in Algorithm 4.3.4.

Algorithm 4.3.4 LRF Rosenbrock of second order

Require: $A \in \mathbb{R}^{n\times n}$, B, C, Z_0 satisfying (4.49), $t \in [a,b]$, and step size h.
Ensure: $(\hat{T}_i, \tilde{T}_i, t_i)$ such that $X_i \approx Z_0 Z_0^T + h(\hat{T}_i\hat{T}_i^T - \tilde{T}_i\tilde{T}_i^T)$.

1: $t_0 = a$.
2: $\tilde{T}_0 = 0$.
3: $\hat{T}_0 = 0$.
4: **for** $k = 0$ to $\lceil \frac{b-a}{h} \rceil$ **do**
5: $F_k = \big[\, (Z_0(Z_0^T B))^T,\ h(\hat{T}_k(\hat{T}_k^T B))^T,\ (\tilde{T}_k(\tilde{T}_k^T B))^T \,\big]$.
6: $\hat{A}_k = \gamma h(A - BF_k) + \frac{1}{2}I$.
7: Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$ with respect to the matrix $\hat{A}_k$.
8: $U_k = \big[\, M_0,\ W_0,\ \sqrt{h}\hat{M}_k,\ \sqrt{2}h\hat{W}_k,\ \sqrt{h}\tilde{L}_k,\ (\sqrt{2h^2+h})\tilde{W}_k,\ \sqrt{h}(W_0 + \hat{W}_k)\big]$.
9: Compute $\tilde{T}_1^k$ by Algorithm 4.2.3 such that the low rank factor product $\tilde{T}_1^k(\tilde{T}_1^k)^T$ approximates the solution of $\hat{A}_k^T\tilde{K}_1 + \tilde{K}_1\hat{A}_k = -U_kU_k^T$.
10: $N_k = \big[\, C^T,\ L_0,\ \sqrt{h}\hat{L}_k,\ \sqrt{h}\tilde{M}_k,\ \sqrt{h}\hat{W}_k,\ \sqrt{h}(W_0 + \tilde{W}_k),\ \sqrt{h}(\hat{W}_k + \tilde{W}_k)\big]$.
11: Compute $\hat{T}_1^k$ by Algorithm 4.2.3 such that the low rank factor product $\hat{T}_1^k(\hat{T}_1^k)^T$ approximates the solution of $\hat{A}_k^T\hat{K}_1 + \hat{K}_1\hat{A}_k = -N_kN_k^T$.
12: $U_k^2 = \big[\, \sqrt{h}\tilde{L}_k^1,\ \sqrt{h}\hat{M}_k^1,\ \sqrt{2}\tilde{T}_1^k,\ \sqrt{2h^2+h}\tilde{W}_k^1,\ \sqrt{2}h\hat{W}_k^1,\ \sqrt{h}(W_0 + \hat{W}_k^1),$
$h(\tilde{W}_k + \tilde{W}_k^1),\ h(\hat{W}_k + \hat{W}_k^1)\,\big]$.
13: Compute $\tilde{T}_2^k$ by Algorithm 4.2.3 such that the low rank factor product $\tilde{T}_2^k(\tilde{T}_2^k)^T$ approximates the solution of $\hat{A}_k^T\tilde{K}_{21} + \tilde{K}_{21}\hat{A}_k = -U_k^2(U_k^2)^T$.
14: $N_k^2 = \big[\, \sqrt{h}\hat{L}_k^1,\ \sqrt{h}\tilde{M}_k^1,\ \sqrt{2}\hat{T}_1^k,\ h(\hat{W}_k^1 + \tilde{W}_k^1),\ \sqrt{h}\hat{W}_k^1,\ \sqrt{h}(W_0 + \tilde{W}_k^1),$
$h(\hat{W}_k + \tilde{W}_k^1),\ h(\tilde{W}_k + \hat{W}_k^1)\,\big]$.
15: Compute $\hat{T}_2^k$ by Algorithm 4.2.3 such that the low rank factor product $\hat{T}_2^k(\hat{T}_2^k)^T$ approximates the solution of $\hat{A}_k^T\hat{K}_{21} + \hat{K}_{21}\hat{A}_k = -N_k^2(N_k^2)^T$.
16: $\tilde{T}_{k+1} = \big[\, \sqrt{2}\tilde{T}_1^k,\ \tilde{T}_k,\ \sqrt{\frac{1}{2}}\tilde{T}_2^k \,\big]$.
17: $\hat{T}_{k+1} = \big[\, \sqrt{2}\hat{T}_1^k,\ \hat{T}_k,\ \sqrt{\frac{1}{2}}\tilde{T}_2^k \,\big]$.
18: $t_{k+1} = t_k + h$.
19: **end for**

Remark 4.3.2 *Steps 10. and 12. as well as 14. and 16. of Algorithm 4.3.4 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.*

Remark 4.3.3 *In the special case of the Rosenbrock method of order two, for autonomous DREs, in which the parameter γ is chosen as 1 ((4.45)–(4.47)), the iteration simplifies considerably solving the second Lyapunov equation (4.47). This results in Algorithm 4.3.5.*

Algorithm 4.3.5 LRF Rosenbrock of second order for $\gamma = 1$

Require: $A \in \mathbb{R}^{n\times n}$, B, C, Z_0 satisfying (4.49), $t \in [a, b]$, and step size h.
Ensure: $(\hat{T}_i, \tilde{T}_i, t_i)$ such that $X_i \approx Z_0 Z_0^T + h(\hat{T}_i\hat{T}_i^T - \tilde{T}_i\tilde{T}_i^T)$.

$t_0 = a$.
$\tilde{T}_0 = 0$.
$\hat{T}_0 = 0$.
for $k = 0$ to $\lceil \frac{b-a}{h} \rceil$ **do**
  $F_k = \big[\, (Z_0(Z_0^T B))^T,\ h(\hat{T}_k(\hat{T}_k^T B))^T,\ (\tilde{T}_k(\tilde{T}_k^T B))^T \,\big]$.
  $\hat{A}_k = h(A - BF_k) + \frac{1}{2}I$.
  Determine (sub)optimal ADI shift parameters $p_1, p_2, \ldots$ with respect to the matrix $\hat{A}_k$.
  $U_k = \big[\, M_0,\ W_0,\ \sqrt{h}\hat{M}_k,\ \sqrt{2}h\hat{W}_k,\ \sqrt{h}\tilde{L}_k,\ (\sqrt{2h^2 + h})\tilde{W}_k,\ \sqrt{h}(W_0 + \hat{W}_k)\big]$.
  Compute $\tilde{T}_1^k$ by Algorithm 4.2.3 such that the low rank factor product $\tilde{T}_1^k(\tilde{T}_1^k)^T$ approximates the solution of $\hat{A}_k^T\tilde{K}_1 + \tilde{K}_1\hat{A}_k = -U_kU_k^T$.
  $N_k = \big[\, C^T,\ L_0,\ \sqrt{h}\hat{L}_k,\ \sqrt{h}\tilde{M}_k,\ \sqrt{h}\hat{W}_k,\ \sqrt{h}(W_0 + \tilde{W}_k),\ \sqrt{h}(\hat{W}_k + \tilde{W}_k)\big]$.
  Compute $\hat{T}_1^k$ by Algorithm 4.2.3 such that the low rank factor product $\hat{T}_1^k(\hat{T}_1^k)^T$ approximates the solution of $\hat{A}_k^T\hat{K}_1 + \hat{K}_1\hat{A}_k = -N_kN_k^T$.
  $U_k^2 = \big[\, \sqrt{2}h\hat{W}_k^1,\ \sqrt{2h}\tilde{W}_k^1,\ \tilde{T}_1^k \,\big]$.
  Compute $\tilde{T}_2^k$ by Algorithm 4.2.3 such that the low rank factor product $\tilde{T}_2^k(\tilde{T}_2^k)^T$ approximates the solution of $\hat{A}_k^T\tilde{K}_2 + \tilde{K}_2\hat{A}_k = -U_k^2(U_k^2)^T$.
  $N_k^2 = \big[\, h(\hat{W}_k^1 + \tilde{W}_k^1),\ \hat{T}_1^k \,\big]$.
  Compute $\hat{T}_2^k$ by Algorithm 4.2.3 such that the low rank factor product $\hat{T}_2^k(\hat{T}_2^k)^T$ approximates the solution of $\hat{A}_k^T\hat{K}_2 + \hat{K}_2\hat{A}_k = -N_k^2(N_k^2)^T$.
  $\tilde{T}_{k+1} = \big[\, \sqrt{\frac{3}{2}}\tilde{T}_1^k,\ \tilde{T}_k,\ \sqrt{\frac{1}{2}}\tilde{T}_2^k \,\big]$.
  $\hat{T}_{k+1} = \big[\, \sqrt{\frac{3}{2}}\hat{T}_1^k,\ \hat{T}_k,\ \sqrt{\frac{1}{2}}\tilde{T}_2^k \,\big]$.
  $t_{k+1} = t_k + h$.
end for

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 by proving that the right hand side of (4.33) and (4.39) respectively, can be expressed as a low rank matrix product. In fact, we just need to prove that F_{t_k} can be represented as a low rank matrix product combination.
If we approximate the derivatives involved in F_{t_k} using central differences as:

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

(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

$$\begin{aligned} F_{t_k} \quad \approx \tfrac{1}{h}\Big[\quad & (Q_k - Q_{k-1}) + hA_k^T F(X_k) + (A_k^T - A_{k-1}^T)X_k \\ & + hF(X_k)A_k + X_k(A_k - A_{k-1}) - hF(X_k)S_k X_k \\ & - X_k(S_k - S_{k-1})X_k - hX_k S_k F(X_k) \Big]. \end{aligned} \tag{4.60}$$

By (4.58) we know that $F(X_k)$ can be expressed as a combination of low rank factor matrix products, then by several computations similar to (4.55) and rearranging terms we can obtain a low rank matrix representation of (4.60). Therefore the Rosenbrock methods for non-autonomous DREs reviewed in Section 4.3.3 can be formulated using low rank factors.

4.4 The ADI parameter selection problem

The alternating direction implicit (ADI) iteration was introduced in [90] as a method for solving elliptic and parabolic difference equations.
Let $A \in \mathbb{R}^{n\times n}$ be a real symmetric positive definite (SPD) and let $s \in \mathbb{R}^n$ be known. We can apply ADI iteration to solve

$$Au = s,$$

when A can be expressed as the sum of matrices H and V for which the linear systems

$$\begin{aligned} (H + pI)v &= r, \\ (V + pI)w &= t \end{aligned}$$

admit an efficient solution. Here p is a suitable chosen parameter and r, t are known.
If H and V are SPD, then there exist positive parameters p_j for which the two-sweep iteration defined by

$$\begin{aligned} (H + p_j I)u_{(j-1)/2} &= (p_j I - V)u_{j-1} + s, \\ (V + p_j I)u_j &= (p_j I - H)u_{j-1/2} + s \end{aligned} \tag{4.61}$$

for $j = 1, 2, \ldots$ converges. If the shift parameters p_j are chosen appropriately, then the convergence rate is superlinear, but convergence rates can be ensured only when matrices H and V commute. In the noncommutative case the ADI iteration is not competitive with other methods. This section is essentially contained in [22].

4.4.1 Introduction

We consider a Lyapunov equation of the form

$$F^T Y + YF = -WW^T \tag{4.62}$$

with stable F, (4.62) is a model ADI problem. The model condition that the component matrices commute is retained. It can be seen when one recognizes that this is equivalent to a linear operator $\mathcal{M}$ mapping Y into $-WW^T$ where $\mathcal{M}$ is the sum of commuting operators: premultiplication of Y by F^T and posmultiplication by F.
Applying the ADI iteration (4.61) to (4.62) yields,

$$\begin{array}{rcl} (F^T + p_j I)Y_{(j-1)/2} & = & -WW^T - Y_{j-1}(F - p_j I), \\ (F^T + \overline{p_j} I)Y_j^T & = & -WW^T - Y_{(j-1)/2}(F - \overline{p}_j I), \end{array} \tag{4.63}$$

where $\overline{p}$ denotes the complex conjugate of $p \in \mathbb{C}$. The matrix $Y_{(j-1)/2}$ is in general not symmetric after the first sweep of each iteration, but the result of the double sweep is symmetric.
Practical experience shows that it is crucial to have good shift parameters to get fast convergence in the ADI process. The error in iterate j is given by $e_j = R_j e_{j-1}$, where

$$R_j := (F + p_j I)^{-1}(F^T - p_j I)(F^T + p_j I)^{-1}(F - p_j I).$$

Thus the error after J iterations satisfies

$$e_J = G_J e_0, \quad G_J := \prod_{j=1}^{J} R_j.$$

Due to the fact that G_J is symmetric,

$$||e_J|| \leq \rho(G_J)||e_0||, \quad \rho(G_J) = k(\mathbf{p})^2,$$

where $\mathbf{p} = \{p_1, p_2, \ldots, p_J\}$ and

$$k(\mathbf{p}) = \max_{\lambda \in \sigma(F)} \left| \prod_{j=1}^{J} \frac{(p_j - \lambda)}{(p_j + \lambda)} \right|. \tag{4.64}$$

By this the ADI parameters are chosen in order to minimize $\rho(G_J)$ which leads to the rational minimax problem

$$\min_{\{p_j \in \mathbb{R}: j=1,\ldots,J\}} k(\mathbf{p}) \tag{4.65}$$

for the shift parameters p_j, see e.g. [115]. This minimization problem is also known as the rational Zolotarev problem since, in the real case (i.e. $\sigma(F) \subset \mathbb{R}$) it is equivalent to the third of four approximation problems solved by Zolotarev in the 19th century, see [79]. For a complete historical overview see [111].

4.4.2 Review of existing parameter selection methods

Many procedures for constructing optimal or suboptimal shift parameters have been proposed in the literature [64, 92, 108, 115]. Most of the approaches cover the spectrum of F by a domain $\Omega \subset \mathbb{C}_-$ and solve (4.65) with respect to Ω instead of $\sigma(F)$. In general one must choose among the various approaches to find effective ADI iteration parameters for specific problems. One could even consider sophisticated algorithms like the one proposed by Istace and Thiran [64] in which the authors use numerical techniques for nonlinear optimization problems to determine optimal parameters. However, it is important to take care that the time spent in computing parameters does not outweigh the convergence improvement derived therefrom.

Wachspress et al. [115] compute the optimum parameters when the spectrum of the matrix F is real or, in the complex case, if the spectrum of F can be embedded in an elliptic functions region, which often occurs in practice. These parameters may be chosen real even if the spectrum is complex as long as the imaginary parts of the eigenvalues are *small* compared to their real parts (see [85, 115] for details). The method applied by Wachspress in the complex case is similar to the technique of embedding the spectrum into an ellipse and then using Chebyshev polynomials. In case that the spectrum is not well represented by the elliptic functions region a more general development by Starke [108] describes how generalized Leja points yield asymptotically optimal iteration parameters. Finally, an inexpensive heuristic procedure for determining ADI shift parameters, which often works well in practice, was proposed by Penzl [92]. We will summarize these approaches here.

Leja Points. Gonchar [55] characterizes the general minimax problem and shows how asymptotically optimal parameters can be obtained with generalized Leja or Fejér points. Starke [107] applies this theory to the ADI minimax problem (4.65). The generalized Leja points are defined as follows. Given $\varphi_j \in E$ and $\psi_j \in F$ arbitrarily, E, F subsets of $\mathbb{C}$, for $j = 1, 2, \ldots$, the new points $\varphi_{j+1} \in E$ and $\psi_{j+1} \in F$ are chosen recursively in such a way that, with

$$r_j(z) = \prod_{i=1}^{j} \frac{z - \varphi_j}{z - \psi_j} \tag{4.66}$$

the two conditions

$$\max_{x \in E} |r_j(z)| = |r_j(\varphi_{j+1})|, \qquad \max_{x \in F} |r_j(z)| = |r_j(\psi_{j+1})|, \tag{4.67}$$

are fullfilled. Bagby [10] shows that the rational functions r_j obtained by this procedure are asymptotically minimal for the rational Zolotarev problem.

Starke considers a general ADI iteration, so for ADI applied to the Lyapunov equation (4.63) the generalized Leja points will be defined as follows:

Given $p_0 \in E$, E is a complex subset such that $\sigma(F) \subset E$, for $j = 1, 2, \ldots,$ the new points $p_j \in E$ are chosen recursively in such a way that, with

$$r_j(z) = \prod_{i=1}^{j} \frac{z - p_j}{z + p_j} \tag{4.68}$$

the condition

$$\max_{x \in E} |r_j(z)| = |r_j(p_{j+1})| \tag{4.69}$$

holds. The generalized Leja points can be determined numerically for a large class of boundary curves ∂E. When relatively few iterations are needed to attain the prescribed accuracy, the Leja points may be poor. Moreover their computation can be quite time consuming when the number of Leja points generated is large, since the computation gets more and more expensive the more prior Leja points are already calculated.

Optimal parameters. In this section we will briefly summarize the parameter selection procedure given in [115].

Define the spectral bounds a, b and a sector angle α for the matrix F as

$$a = \min_i(Re\{\lambda_i\}), \quad b = \max_i(Re\{\lambda_i\}), \quad \alpha = \tan^{-1} \max_i \left| \frac{Im\{\lambda_i\}}{Re\{\lambda_i\}} \right|, \tag{4.70}$$

where $\lambda_1, \ldots, \lambda_n$ are eigenvalues of $-F$. It is assumed that the spectrum of $-F$ lies inside the elliptic functions region determined by a, b, α, as defined in [115]. Let

$$\cos^2 \beta = \frac{2}{1 + \frac{1}{2}\left(\frac{a}{b} + \frac{b}{a}\right)}, \quad m = \frac{2\cos^2 \alpha}{\cos^2 \beta} - 1. \tag{4.71}$$

If $\alpha < \beta$, then $m \geq 1$ and the parameters are real. We define

$$k_1 = \frac{1}{m + \sqrt{m^2 - 1}}, \quad k = \sqrt{1 - {k_1}^2}. \tag{4.72}$$

Define the elliptic integrals K and v via

$$F[\psi, k] = \int_0^{\psi} \frac{dx}{\sqrt{1 - k^2 \sin^2 x}}, \tag{4.73}$$

as

$$K = K(k) = F\left[\frac{\pi}{2}, k\right], \quad v = F\left[\sin^{-1}\sqrt{\frac{a}{bk_1}}, k_1\right], \tag{4.74}$$

where F is the incomplete elliptic integral of the first kind, k is its modulus and ψ is its amplitude.

The number of the ADI iterations required to achieve $k(\mathbf{p})^2 \leq \epsilon$ is $J = \lceil \frac{K}{2v\pi} \log \frac{4}{\epsilon} \rceil$, and the ADI parameters are given by

$$p_j = -\sqrt{\frac{ab}{k_1}} \operatorname{dn}\left[\frac{(2j-1)K}{2J}, k\right], \quad j = 1, 2, \ldots, J, \tag{4.75}$$

where $\operatorname{dn}(u, k)$ is the elliptic function (see [3]).

If $m < 1$, the parameters are complex. We define the dual elliptic spectrum,

$$a' = \tan\left(\frac{\pi}{4} - \frac{\alpha}{2}\right), \quad b' = \frac{1}{a'}, \quad \alpha' = \beta.$$

Substituting a' in (4.71), we find that

$$\beta' = \alpha, \quad m' = \frac{2\cos^2\beta}{\cos^2\alpha} - 1.$$

By construction, m' must now be greater than 1. Therefore we may compute the optimum real parameters p'_j for the dual problem. The corresponding complex parameters for the actual spectrum can then be computed from:

$$\cos\alpha_j = \frac{2}{p'_j + \frac{1}{p'_j}}, \tag{4.76}$$

and for $j = 1, 2, \ldots, \lceil \frac{1+J}{2} \rceil$

$$p_{2j-1} = \sqrt{ab}\exp[\imath\alpha_j], \quad p_{2j} = \sqrt{ab}\exp[-\imath\alpha_j]. \tag{4.77}$$

Heuristic parameters. The bounds needed to compute optimal parameters are too expensive to be computed exactly in case of large scale systems because they need the knowledge of the whole spectrum of F. In fact, this computation would be more expensive than the application of the ADI method itself.

An alternative was proposed by Penzl in [92]. He presents a heuristic procedure which determines suboptimal parameters based on the idea of replacing $\sigma(F)$ by an approximation $\mathcal{R}$ of the spectrum in (4.65). Specifically, $\sigma(F)$ is approximated using the Ritz values computed by the Arnoldi process (or any other large scale eigensolver). Due to the fact that the Ritz values tend to be located near the largest magnitude eigenvalues, the inverses of the Ritz values related to F^{-1} are also computed to get an approximation of the smallest magnitude eigenvalues of F yielding a better approximation of $\sigma(F)$. The suboptimal parameters $\mathcal{P} = \{p_1, \ldots, p_k\}$ are chosen among the elements of this approximation because the function

$$s_{\mathcal{P}}(t) = \frac{|(t-p_1)\ldots(t-p_k)|}{|(t+p_1)\ldots(t+p_k)|}$$

becomes small over $\sigma(F)$ if there is one of the shifts p_j in the neighborhood of each eigenvalue. The procedure determines the parameters as follows. First,

the element $p_j \in \mathcal{R}$ which minimizes the function $s_{\{p_j\}}$ over $\mathcal{R}$ is chosen. The set $\mathcal{P}$ is initialized by either $\{p_j\}$ or the pair of complex conjugates $\{p_j, \bar{p}_j\}$. Now $\mathcal{P}$ is successively enlarged by the elements or pairs of elements of $\mathcal{R}$, for which the maximum of the current $s_{\mathcal{P}}$ is attained. Doing this the elements of $\mathcal{R}$ giving the largest contributions to the value of $s_{\mathcal{P}}$ are successively canceled out. Therefore the resulting $s_{\mathcal{P}}$ is nonzero only in the elements of $\mathcal{R}$ where its value is comparably small anyway. In this sense (4.65) is solved heuristicly.

Discussion. We are searching for a parameter set for the ADI method applied to a control problem, where in the PDE constraint (1.1) the diffusive part is dominating the reaction or convection terms, respectively. Thus the resulting operator has a spectrum with only moderately large imaginary components compared to the real parts. In these problems the Wachspress approach should always be applicable and lead to real shift parameters in many cases. In problems, where the reactive and convective terms are absent, i.e. we are considering a plain heat equation and therefore the spectrum is part of the real axis, the Wachspress parameters are proven to be optimal. The heuristics proposed by Penzl is more expensive to compute there and Starke notes in [107], that the generalized Leja approach will not be competitive here since it is only asymptotically optimal. For the complex spectra case common strategies to determine the generalized Leja points generalize the idea of enclosing the spectrum by a polygonal domain, where the starting roots are placed in the corners. So one needs quite exact information about the shape of the spectrum there. In practice this would require to be able to compute the eigenvalues with largest imaginary parts already for a simple rectangular enclosure of the spectrum. Since this still does not work reliably, we decided to avoid the comparison with that approach in this publication, although it might proof useful in cases where the Wachspress parameters are no longer applicable or one knows some a-priori information on the spectrum.

4.4.3 Suboptimal parameter computation

In this section we discuss our new contribution to the parameter selection problem. The idea is to avoid the problems of the methods reviewed in the previous section and on the other hand combine their advantages.

Since the important information that we need to know for the Wachspress approach is the outer shape of the spectrum of the matrix F, we will describe an algorithm approximating the outer spectrum. With this approximation the input parameters a, b and α for the Wachspress method are determined and the optimal parameters for the approximated spectrum are computed. Obviously, these parameters have to be considered suboptimal for the original problem, but if we can approximate the outer spectrum at a similar cost to that of the heuristic parameter choice we end up with a method giving nearly optimal parameters at a drastically reduced computational cost compared to the optimal parameters.

In the following we discuss the main computational steps in Algorithm 4.4.1.

Algorithm 4.4.1 approximate optimal ADI parameter computation

Require: F Hurwitz stable

if $\sigma(F) \subset \mathbb{R}$ **then**
  Compute the spectral bounds and set $a = \min \sigma(-F)$ and $b = \max \sigma(-F)$,
  $k_1 = \frac{a}{b}$, $k = \sqrt{1 - k_1^2}$,
  $K = F(\frac{\pi}{2}, k)$, $v = F(\frac{\pi}{2}, k_1)$.
  Compute J and the parameters according to (4.75).
else
  Compute $\tilde{a} = \min \mathrm{Re}\,(\sigma(-F))$, $\tilde{b} = \max \mathrm{Re}\,(\sigma(-F))$ and $c = \frac{\tilde{a}+\tilde{b}}{2}$.
  Compute l largest magnitude eigenvalues $\hat{\lambda}_i$ for the shifted matrix $-F+cI$ by an Arnoldi process or alike.
  Shift these Eigenvalues back, i.e. $\tilde{\lambda}_i = \hat{\lambda}_i + c$.
  Compute a, b and α from the $\tilde{\lambda}_i$ like in (4.70).
  if $m \geq 1$ in (4.71) **then**
    Compute the parameters by (4.71)–(4.75).
  else {The ADI parameters are complex in this case}
    Compute the dual variables.
    Compute the parameters for the dual variables by (4.71)–(4.75).
    Use (4.76) and (4.77) to get the complex shifts.
  end if
end if

Real spectra In the case where the spectrum is real we can simply compute the upper and lower bounds of the spectrum by an Arnoldi or Lanczos process and enter the Wachspress computation with these values for a and b, and set $\alpha = 0$, i.e., we only have to compute two complete elliptic integrals by an arithmetic geometric mean process. This is very cheap since it is a quadratically converging scalar computation (see below).

Complex spectra For complex spectra we introduce an additional shifting step to be able to apply the Arnoldi process more efficiently. Since we are dealing with stable systems[2], we compute the largest magnitude and smallest magnitude eigenvalues and use the arithmetic mean of their real parts as a horizontal shift, such that the spectrum is centered around the origin. Now Arnoldi's method is applied to the shifted spectrum, to compute a number of largest magnitude eigenvalues. These will now automatically include the smallest magnitude eigenvalues of the original system after shifting back. So we can avoid extensive application of the Arnoldi method to the inverse of F. We only need it to get a rough approximation of the smallest magnitude eigenvalue to determine $\tilde{a}$ and $\tilde{b}$ for the shifting step.

The number of eigenvalues we compute can be seen as a tuning parameter

[2]Note that the Newton-ADI-iteration assumes that we know a stabilizing initial feedback, or the system is stable itself

here. The more eigenvalues we compute, the better the approximation of the shape of the spectrum is and the closer we get to the exact a, b and α, but obviously the computation becomes more and more expensive. Especially the dimension of the Krylov subspaces is rising with the number of parameters requested and with it the memory consumption in the Arnoldi process. But in cases where the spectrum is filling a rectangle or an egg-like shape, a few eigenvalues are sufficient here (compare Section 4.4.4).

A drawback of this method can be that in case of small (compared to the real parts) imaginary parts of the eigenvalues, one may need a large number of eigenvalue approximations to find the ones with largest imaginary parts, which are crucial to determine α accurately. On the other hand in that case the spectrum is *almost* real and therefore it will be sufficient to compute the parameters for the approximate real spectrum in most applications.

Computation of the elliptic integrals The new as well as the Wachspress parameter algorithms require the computation of certain elliptic integrals presented in (4.73). These are equivalent to the integral

$$F[\psi, k] = \int_0^{\psi} \frac{dx}{\sqrt{(1-k^2)\sin^2 x + \cos^2 x}} = \int_0^{\psi} \frac{dx}{\sqrt{(k_1^2)\sin^2 x + \cos^2 x}}. \quad (4.78)$$

In the case of real spectra, $\psi = \frac{\pi}{2}$ and $F[\frac{\pi}{2}, k]$ is a complete elliptic integral of the form

$$I(a,b) = \int_0^{\frac{\pi}{2}} \frac{dx}{\sqrt{a^2\cos^2 x + b^2 \sin^2 x}}$$

and $I(a,b) = \frac{\pi}{2M(a,b)}$, where $M(a,b)$ is the arithmetic geometric mean of a and b. The proof for the quadratic convergence of the arithmetic geometric mean process is given in many textbooks (e.g., [110]).

For incomplete elliptic integrals, i.e., the case $\psi < \frac{\pi}{2}$, an additional Landen's transformation has to be performed. Here, first the arithmetic geometric mean is computed as above, then a descending Landen's transformation is applied (see [3, Chapter 17]), which comes in at the cost of a number of scalar tangent computations equal to the number of iteration steps taken in the arithmetic geometric mean process above.

The value of the elliptic function dn from equation (4.75) is also computed by an arithmetic geometric mean process (see [3, Chapter 16]).

To summarize the advantages of the proposed method we can say:

- We compute real shift parameters even in case of many complex spectra, where the heuristic method would compute complex ones. This results in a significantly cheaper ADI iteration considering memory consumption and computational effort, since complex computations are avoided.
- We have to compute less Ritz values compared to the heuristic method, reducing the time spent in the computational overhead for the acceleration of the ADI method.

- We compute a good approximation of the Wachspress parameters at a drastically reduced computational cost compared to their exact computation.

4.4.4 Numerical results

For the numerical tests we used the LyaPack[3] software package [94]. A test program similar to `demo_r1` from the LyaPack examples is used for the computation, where the ADI parameter selection is switched between the methods described in the previous sections. We are here concentrating on the case where the ADI shift parameters can be chosen real.

FDM semi-discretized diffusion-convection-reaction equation. Here we consider the finite difference semi-discretized partial differential equation

$$\frac{\partial \mathbf{x}}{\partial t} - \Delta \mathbf{x} - \begin{bmatrix} 20 \\ 0 \end{bmatrix} . \nabla \mathbf{x} + 180\mathbf{x} = \mathbf{f}(\xi)\mathbf{u}(t), \tag{4.79}$$

where $\mathbf{x}$ is a function of time t, vertical position ξ_1 and horizontal position ξ_2 on the square with opposite corners $(0, 0)$ and $(1, 1)$. The example is taken from the SLICOT collection of benchmark examples for model reduction of linear time-invariant dynamical systems (see [36, Section 2.7] for details). It is given in semi-discretized state space model representation:

$$\dot{x} = Ax + Bu, \quad y = Cx. \tag{4.80}$$

The matrices A, B, C for this system can be found on the NICONET web site[4].

Figure 4.2 (a),(b) show the spectrum and sparsity pattern of the system matrix A. The iteration history, i.e., the numbers of ADI steps in each step of Newton's method are plotted in Figure 4.2 (c). There we can see that in fact the semi-optimal parameters work exactly like the optimal ones by the Wachspress approach. This is what we would expect since the rectangular spectrum is an optimal case for our idea, because the parameters a, b and α are exactly (to the accuracy of Arnoldi's method) met here. Note especially that for the heuristic parameters even more outer Newton iterations than for our parameters are required.

FDM semi-discretized heat equation. In this example we tested the parameters for the finite difference semi-discretized heat equation on the unit square $(0, 1) \times (0, 1)$.

$$\frac{\partial \mathbf{x}}{\partial t} - \Delta \mathbf{x} = \mathbf{f}(\xi)\mathbf{u}(t). \tag{4.81}$$

The data is generated by the routines `fdm_2d_matrix` and `fdm_2d_vector` from the examples of the LyaPack package. Details on the generation of test problems

[3] available from: `http://www.netlib.org/lyapack/` or `http://www.tu-chemnitz.de/sfb393/lyapack/`

[4] `http://www.icm.tu-bs.de/NICONET/benchmodred.html`

can be found in the documentation of these routines (comments and MATLAB help). Since the differential operator is symmetric here, the matrix A is symmetric and its spectrum is real in this case. Hence $\alpha = 0$ and for the Wachspress parameters only the largest magnitude and smallest magnitude eigenvalues have to be found to determine a and b. That means we only need to compute two Ritz values by the Arnoldi (which here is in fact a Lanczos process because of symmetry) process compared to about 30 (which seems to be an adequate number of shifts) for the heuristic approach. We used a test example with 400 unknowns here to still be able to compute the complete spectrum using `eig` for comparison.

In Figure 4.3 we plotted the sparsity pattern of A and the iteration history for the solution of the corresponding ARE. We can see (Figure 4.3 (b)) that iteration numbers only differ very slightly. Hence we can choose quite independently which parameters to use. Since the Wachspress approach needs a good approximation of the smallest magnitude eigenvalue it might be a good idea to choose the heuristic parameters here (even though they are much more expensive to compute) if the smallest magnitude eigenvalue is known to be close to the origin (e.g. in case of finite element discretizations with fine meshes).

FEM semi-discretized convection-diffusion equation. The last example is a system appearing in the optimal heating/cooling of a fluid flow in a tube. An application is the temperature regulation of certain reagent inflows in chemical reactors. The model equations are:

$$\begin{array}{rcll} \frac{\partial \mathbf{x}}{\partial t} - \alpha\Delta\mathbf{x} + \mathbf{v}\cdot\nabla\mathbf{x} & = & 0 & \text{in } \Omega \\ \mathbf{x} & = & \mathbf{x}_0 & \text{on } \Gamma_{in} \\ \frac{\partial \mathbf{x}}{\partial n} & = & \sigma(\mathbf{u}-\mathbf{x}) & \text{on } \Gamma_{heat1} \cup \Gamma_{heat2} \\ \frac{\partial \mathbf{x}}{\partial n} & = & 0 & \text{on } \Gamma_{out}. \end{array} \tag{4.82}$$

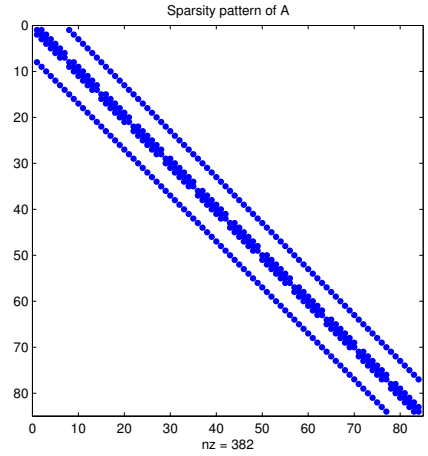
Here Ω is the rectangular domain shown in Figure 4.4 (a). The inflow Γ_{in} is at the left part of the boundary and the outflow Γ_{out} the right one. The control is applied via the upper and lower boundaries. We can restrict ourselves to this 2d-domain assuming rotational symmetry, i.e., non-turbulent diffusion dominated flows. The test matrices have been created using the COMSOL Multiphysics software and $\alpha = 0.06$, resulting in the Eigenvalue and shift distributions shown in Figure 4.4 (b).

Since a finite element discretization in space has been applied here, the semi-discrete model is of the form

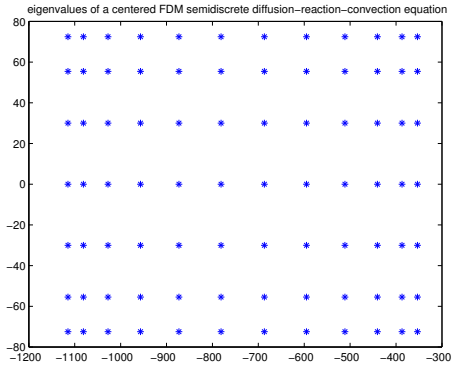
$$\begin{array}{rcl} M\dot{x} & = & \tilde{A}x + \tilde{B}u \\ y & = & \tilde{C}x. \end{array} \tag{4.83}$$

This is transformed into a standard system (4.80) by decomposing M into $M = M_L M_U$ where $M_L = M_U^T$ since M is symmetric. Then defining $\tilde{x} := M_U x$, $A := M_L^{-1}\tilde{A}M_U^{-1}$, $B := M_L^{-1}$ and $C := \tilde{C}M_U^{-1}$ (without computing any of the inverses explicitly in the code) we end up with a standard system for $\tilde{x}$ having the same inputs u as (4.83).

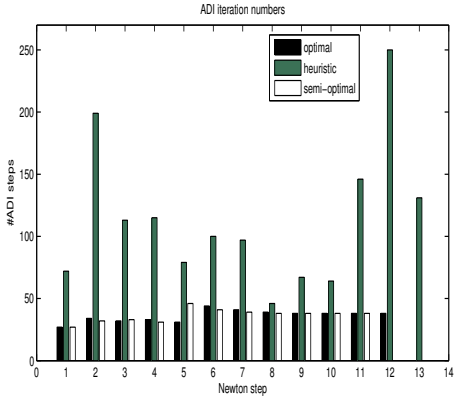
Note, that the heuristic parameters do not appear in the results bar graphics here. This is due to the fact, that the LyaPacksoftware crashed while applying the complex shift computed by the heuristics. Numerical tests where only the real ones of the heuristic parameters where used lead to very poor convergence in the inner loop, which is generally stopped by the maximum iteration number stopping criterion. This resulted in breaking the convergence in the outer Newton loop.



(a)

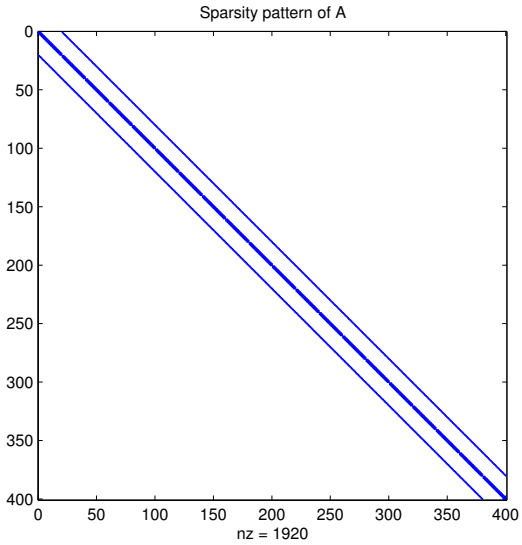


(b)

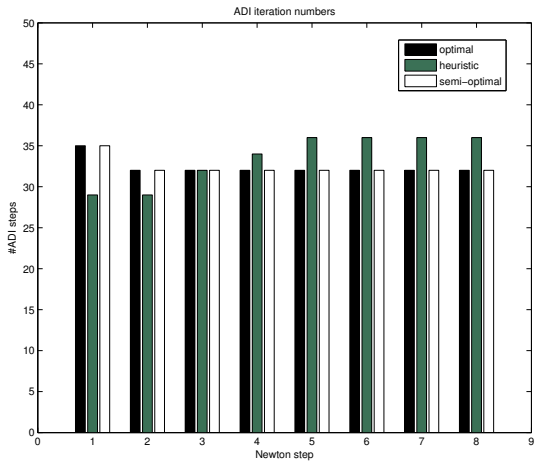


(c)

Figure 4.2: (a) sparsity pattern of the FDM semi-discretized operator for equation (4.79) and (b) its spectrum (c) Iteration history for the Newton ADI method applied to (4.79)

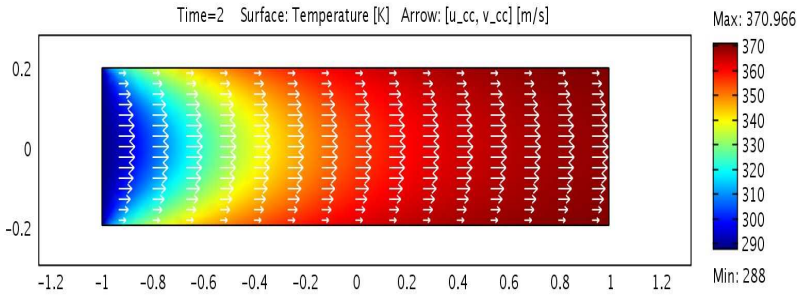


(a)

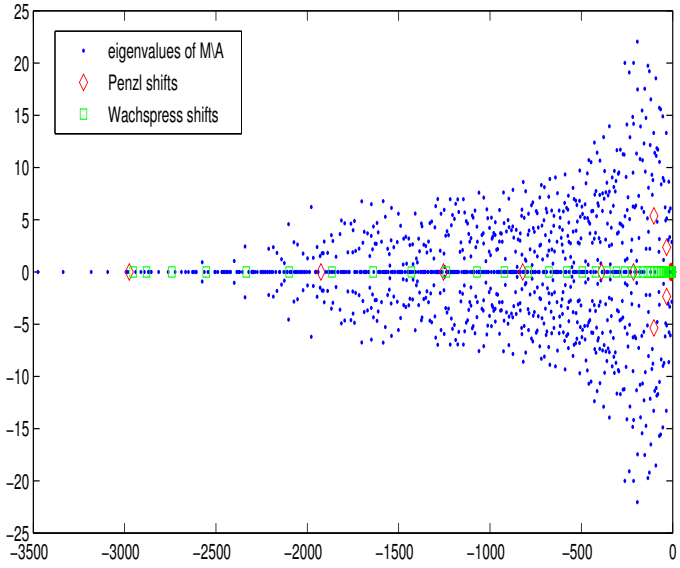


(b)

Figure 4.3: (a) sparsity pattern of the FDM semi-discretized operator for equation (4.81) and (b) Iteration history for the Newton ADI

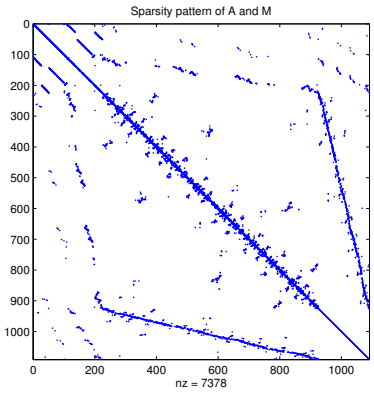


(a)

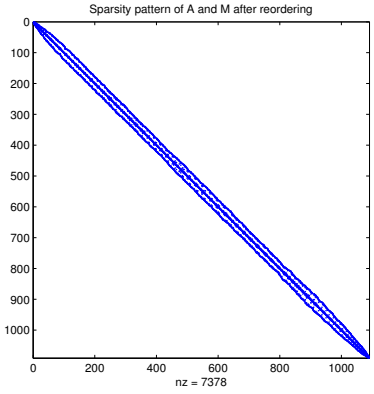


(b)

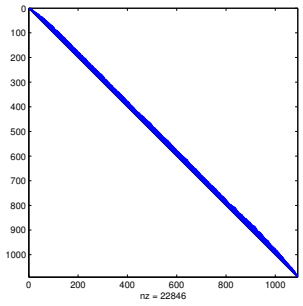
Figure 4.4: (a) A 2d cross-section of the liquid flow in a round tube. (b) Eigenvalue and shift parameter distributions.



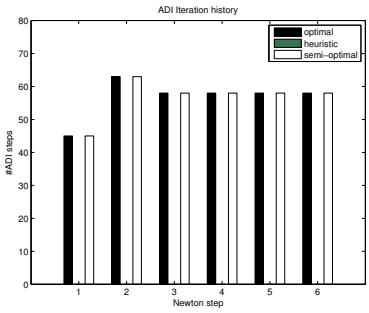
(a)



(b)



(c)



(d)

Figure 4.5: (a) sparsity pattern of A and M in (4.83) , (b) sparsity pattern of A and M in (4.83) after reordering for bandwidth reduction, (c) sparsity pattern of the Cholesky factor of reordered M and (d) Iteration history for the Newton ADI.

CHAPTER

FIVE

Numerical examples for DREs

In this chapter we present numerical experiments for solving DREs by the BDF and the Rosenbrock methods proposed in Chapter 4. In Section 5.1 we describe the examples in which we tested the efficiency of our algorithms. Then, in Section 5.2 we discuss the behavior of our methods and analyze the accuracy of the solutions computed by the different methods. The performance of the fixed step size methods is shown in Section 5.2.1. Finally, in Section 5.2.2 we discuss the suitability of the variable step size methods for large-scale problems. A general comparison among the methods is presented also. We implemented our codes in MATLAB7.0.4.

5.1 Examples

Let us first consider an example of small dimension to be able to analyze the performance of our methods in every component of the solution. For this example, we vectorize the DRE and compare the efficiency of our methods with the standard stiff ODE MATLAB solver `ode23s`. In Example 2, a DRE is considered whose analytic solution is known and its size can be chosen arbitrarily. These allow us to analyze the error of the methods. Then, as a first approach to the application to control problems, we have considered a DRE where the data come from a linear-quadratic control problem of one-dimensional heat flow. This is a parameter dependent and variable size problem. Finally in Example 4, we modify Example 3 in such a way that it results in a time-varying DRE.
By Remark 2.2.7, the solution of the DRE must converge to the solution of the ARE when the interval of integration increases. As a measure of the good performance of our code, we have plotted this convergence for Examples 1, 2 and 3.

Example 1. Let us consider the DRE

$$\begin{aligned} \dot{X}(t) &= Q + A^T X(t) + X(t)A - X(t)SX(t), \\ X(0) &= X_0, \end{aligned} \tag{5.1}$$

where

$$Q = \begin{bmatrix} 9 & 6 \\ 6 & 4 \end{bmatrix}, \quad A = \begin{bmatrix} 4 & 3 \\ -\frac{9}{2} & -\frac{7}{2} \end{bmatrix}, \quad S = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},$$

and

$$X(0) = \begin{bmatrix} 0.5625 & -0.5625 \\ -0.5625 & 0.5625 \end{bmatrix}.$$

If we do note $X(t) = [x_{ij}(t)]$ $i, j = 1, 2$ and vectorize the DRE this yields the ODE system

$$\begin{aligned} \dot{x}_{11}(t) &= 9 + 8x_{11}(t) - 9x_{12}(t) - x_{11}^2(t) + 2x_{11}(t)x_{12}(t) - x_{12}^2(t), \\ \dot{x}_{12}(t) &= 6 + 3x_{11}(t) - 4.5x_{22}(t) + 0.5x_{12}(t) - x_{12}(t)x_{11}(t) + x_{11}(t)x_{22}(t) \\ &\quad -x_{12}(t)x_{22}(t) + x_{12}^2(t), \\ \dot{x}_{22}(t) &= 4 + 6x_{12}(t) - 7x_{22}(t) - x_{12}^2(t) + 2x_{12}(t)x_{22}(t) - x_{22}^2(t), \end{aligned}$$

due to the symmetry of $X(t)$, $x_{21} = x_{12}$.
Notice that the solution of the associate ARE is

$$X_* = \begin{bmatrix} 9(1+\sqrt{2}) & 6(1+\sqrt{2}) \\ 6(1+\sqrt{2}) & 4(1+\sqrt{2}) \end{bmatrix}.$$

In Figure 5.2, we plot the approximation of each solution component of (5.1) by the BDF methods as well as by the Rosenbrock methods using fixed step size. Methods of the same order are compared in Figure 5.3(a) and 5.3(b). The convergence of the DRE to the associated ARE is plotted (for each solution component) in Figure 5.3(c), 5.3(d), and 5.3(e). We use here a relatively large step size to be able to visualize the behavior of each method.

Example 2. Let us now consider the following symmetric DRE of size n,

$$\begin{aligned} \dot{X}(t) &= -X^2(t) + k^2 I_n, \\ X(t_0) &= X_0 \qquad\qquad t_0 \le t \le T. \end{aligned} \tag{5.2}$$

If X_0 is diagonalizable, i.e, $X_0 = S\Lambda S^{-1}$ with Λ=diag$[\lambda_i]$, then the analytic solution of (5.2) is:

$$X(t) = S\text{diag}\left[\frac{k\sinh kt + \lambda_i \cosh kt}{\cosh kt + \frac{\lambda_i}{k}\sinh kt}\right] S^{-1},$$

refer to [37] for a detailed explanation. Here, we choose

$$X_0 = I_n, \quad k = 3, \quad n = 60.$$

In Figure 5.4 (a), we plot the exact solution component X11 and its approximation, by the BDF methods, (b) and by the Rosenbrock methods. The convergence of the DRE to the associated ARE for this solution component is plotted in Figures (c) and (d). Finally, the number of Newton iterations per step for the second order BDF method (BDF2) is shown in (e), and for the third order BDF method (BDF3) in (f).
The error for the BDF and the Rosenbrock methods is shown in Figure 5.5.
In Figure 5.6 (a), we plot the approximate solution component X11 by variable step size and order BDF methods up to order 3 (BDF123) and by variable step size Rosenbrock method of order two (Ros12) in (b). The behavior of the step sizes is shown in (c) and (d). Finally, the error vs. step size is plotted in (e) and (f). The tolerance to accept or redo the current step was choosen as $Tol = 1e-4$.

Example 3. The data of this example arises in a linear-quadratic control problem of a one-dimensional heat flow. This problem is described in terms of infinite-dimensional operators on a Hilbert space. Using a standard finite element approach based on linear B-splines, a finite-dimensional approximation to the problem may be obtained by the solution of AREs. This example was taken from the SLICOT collection of benchmark examples for continuous-time algebraic Riccati equations (see [1] for details).
By Remark 2.2.7, we consider here the associated DRE

$$\begin{aligned} \dot{X}(t) &= (C^T\tilde{Q})(C^T\tilde{Q})^T + A^TX(t) + X(t)A \\ &\quad -X(t)(BR^{-1}\tilde{R})(BR^{-1}\tilde{R})^TX(t), \end{aligned} \tag{5.3}$$

where the matrices C, $\tilde{Q}$, A, B and $\tilde{R}$ come from the ARE arising in the discretized problem. The initial condition for (5.3) is equal to zero.
If N denotes the number of sampling nodes, then with this approach a system of order $n = N - 1$ is obtained.
The system matrices are given by

$$A = M_N^{-1}K_N, \quad B = M_N^{-1}b_N, \quad R = 1, \quad C = c_N^T, \quad \tilde{Q} = 1,$$

where $K_N \in \mathbb{R}^{n\times n}$ is defined as

$$K_N = -aN \begin{bmatrix} 2 & -1 & 0 & & \dots & 0 \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \vdots \\ & & & -1 & 2 & -1 \\ 0 & \dots & & & -1 & 2 \end{bmatrix},$$

Test	n	a	$b = c$	$[\beta_1, \beta_2]$	$[\gamma_1, \gamma_2]$
1	20	0.05	0.1	[0.1,0.5]	[0.1,0.5]
2	100	0.01	1.0	[0.2,0.3]	[0.2,0.3]
3	200	0.01	1.0	[0.2,0.3]	[0.2,0.3]

Table 5.1: Problem parameters for one-dimensional heat flow.

$M_N \in \mathbb{R}^{n\times n}$ as

$$M_N = \frac{1}{6N}\begin{bmatrix} 4 & 1 & 0 & & \dots & 0 \\ 1 & 4 & 1 & & & \\ & \ddots & \ddots & \ddots & & \vdots \\ & & & 1 & 4 & 1 \\ 0 & \dots & & & 1 & 4 \end{bmatrix},$$

and b_N, $c_N \in \mathbb{R}^{n\times 1}$ are given by

$$\begin{aligned} (b_N)_i &= \int_0^1 \beta(s)\varphi_i^N(s)ds, \quad i = 1, \dots, n, \\ (c_N)_i &= \int_0^1 \gamma(s)\varphi_i^N(s)ds, \quad i = 1, \dots, n, \end{aligned}$$

Here $\{\varphi_i^N\}_{i=1}^n$ is the B-spline basis for the chosen finite-dimensional subspace of the underlying Hilbert space. The functions β, $\gamma \in L^2(0,1)$ are defined by

$$\begin{aligned} \beta(s) &= \begin{cases} b, & s \in [\beta_1, \beta_2] \\ 0, & \text{otherwise} \end{cases}, \\ \gamma(s) &= \begin{cases} c, & s \in [\gamma_1, \gamma_2] \\ 0, & \text{otherwise} \end{cases}. \end{aligned}$$

Besides the system dimension n, the problem has the parameters a, b, c, β_1, β_2, γ_1, and γ_2. The problem parameters chosen here are shown in Table 5.1. Test 2 corresponds to the default values of this benchmark example. Test 3 results in a finer grid for this approximation example.
In Figure 5.7 (a), for Test 1 we plotted the approximation of the solution component X11, corresponding to (5.3), by the BDF methods and by the Rosenbrock methods in (b). The convergence of the DRE to the associated ARE for this solution component is plotted in Figures (c) and (d). Finally, the number of Newton iterations per step for BDF2 (e) and for BDF3 (f) are also plotted.
In Figure 5.8 (a), for Test 2 we plotted the approximation of the solution component X11 by the BDF methods, and for Test 3 the approximation of the solution component X13 in (b). The same pictures are plotted for the Rosenbrock methods in Figures (c) and (d), respectively. Finally, for Test 2 the number of Newton

Test	n	$b = c$	$[\beta_1, \beta_2]$	$[\gamma_1, \gamma_2]$
1	100	1.0	[0.2,0.3]	[0.2,0.3]

Table 5.2: Problem parameters for nonlinear one-dimensional heat flow.

iterations per step by BDF3 is shown in (e) and for Test 3 in (f).
In Figure 5.9 (a), we plotted an approximation of the solution component X11 by BDF123 and by Ros12 for Test 1, and for Test 2 in (b). For Test 1, the error vs. step size by BDF123 is shown in (c) and for Test 2 in (b). The same are plotted for Ros12 in (e) and (f), respectively. The tolerance for accept or redo the current step was choosen as $Tol = 1e-7$.

Example 4. Let us consider the problem of optimal cooling of steel profiles [49, 103, 112]. There, the diffusive part is nonlinear. The linearization is derived by taking means of the material parameters: heat conductivity λ, heat capacity c and density ϱ. It is pointed out in [103] that these parameters are modeled in terms of the temperature by

$$\begin{aligned} \varrho(\theta) &= -0.4553\theta + 7988, \\ \lambda(\theta) &= 0.0127\theta + 14.6, \\ c(\theta) &= 0.1756\theta + 454.4, \end{aligned}$$

where $\theta \in [700, 1000]$, and the nonlinear term (analogous to parameter a in Example 3) is defined as

$$\tilde{a}(\theta) = \frac{\lambda(\theta)}{c(\theta)\varrho(\theta)}.$$

We can see in Figure 5.1 that $\tilde{a}(t)$ is strictly increasing. Based on this, as a first approach to solve nonlinear problems (and therefore time-varying DREs), we analyze Example 3 in the time interval $[0, 15]$, redefining the parameter a as a piecewise constant function of the form

$$a(t) = \begin{cases} 0.008 & \text{if } t \in [0, 3[, \\ 0.0085 & \text{if } t \in [3, 6[, \\ 0.009 & \text{if } t \in [6, 9[, \\ 0.0095 & \text{if } t \in [9, 12[, \\ 0.01 & \text{if } t \in [12, 15]. \end{cases} \tag{5.4}$$

The problem parameters chosen are shown in Table 5.2. In Figure 5.10 (a), we plot the approximate solution component X11 by the BDF methods, including BDF123, and by the Rosenbrock methods, including Ros12, in (b). The step sizes over time for BDF123 is shown in (c) and for Ros12 in (d). Finally, the error vs. step size for BDF123 is plotted in (e) and for Ros12 in (f). The tolerance for accept or redo the current step was choosen as $Tol = 1e-8$.

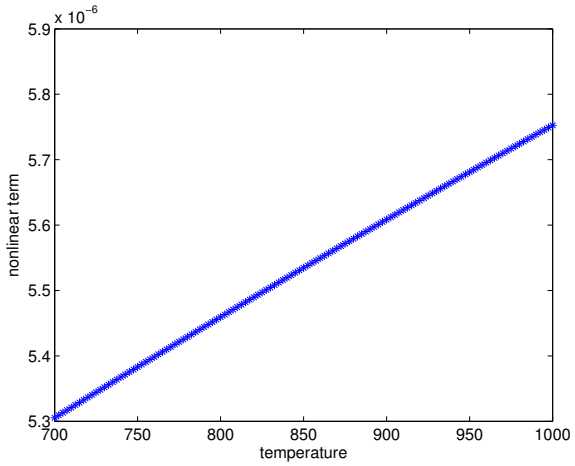


Figure 5.1: Temperature distribution of the nonlinear term $\tilde{a}(t)$.

5.2 Discussion

For the numerical experiments of the methods proposed in Chapter 4 we have considered DREs of moderate size. We restrict ourselves to methods up to order three. Besides the fact that the accuracy demand in the problems we expect to deal with is not high (if the accuracy demand is modest, low order methods are the natural choice), we are interested in large-scale applications where higher order methods are not feasible to apply due to the computational cost and memory requirements.
For Example 1, even though, the DRE seems to be not stiff (at least for the component we plotted, as we can see in Figure 5.2) we compare our methods with the stiff code `ode23s` to make a fair comparison.
In Example 2, the accuracy of the computed solutions of the BDF and the Rosenbrock methods was verified. As we can see in Figure 5.5 the order of the methods are attained for the BDF and the Rosenbrock methods as well.
The convergence of the solution of the DRE to the associated ARE solution is achieved for all approximated solution components that have been analyzed using the different methods we tested.

5.2.1 Fixed step size

As we expected the behavior of the methods of the same order is quite similar for all these examples, the error analysis (Figure 5.5) confirms this fact. From the computational cost point of view, the cost of applying the implicit Euler method (BDF1) is that of solving one ARE per step, and even though we expect to have just a few Newton iterations per step (as we can see form Figure 5.4 (e) and (f), Figure 5.7 (e) and (f), Figure 5.8(e) and (f)) it is more expensive than

the cost of applying the linearly implicit Euler method (Ros1) which a cost of solving one Lyapunov equation per step. We point out that the computational cost of solving the Lyapunov equation involved in every step applying Ros1 is quite similar to the one involved solving the ARE by Newton iteration in BDF1, in fact they can be solved almost at the same cost (remember that the Rosenbrock methods can be interpreted as the application of one Newton step to each stage). So roughly speaking, the application of the Ros1 method is M times cheaper than the application of BDF1, where M is the average number of Newton iterations per step solving the ARE involved in BDF1. On the other hand, for some stiff ODE problems BDF1 behaves better than Ros1 which could occur solving DREs with our approach as well.
This pattern holds for a general comparison between the BDF methods and Rosenbrock methods for small-scale DREs, i.e., in general the Rosenbrock method of order p will be cheaper to compute than the BDF method of order p. This relies on the fact that Rosenbrock methods do not require the added burden of iteration to accomplish the task of solving the implicit equation resulting from the application of the method. However, the more expensive computation resulting from the application of the BDF methods may be rewarded with a better behavior. So, we can not give a general criterion for choosing among these methods. It really depends on the specific application that we are dealing with.
In large-scale problems the situation changes. The cost of solving the Lyapunov equation in each stage of the Rosenbrock method of order p ($p \geq 2$) increase because the low rank factor of the approximating solution is not computed directly to keep working in real arithmetics (instead two low rank factors are computed which approximate this low rank factor, see Section 4.3.4). This makes the algorithm more expensive. However, the L-stable Rosenbrock method of order two which we have been dealing with is capable of integrating with large a priori described step sizes with satisfactory results using moderate accuracies for large-scale ODEs arising from atmospheric dispersion problems, [30]. Thus, this method still could be an option for large-scale problems.
We conclude that in general fixed step size solvers are an option to be considered for large-scale problems.

5.2.2 Variable step size

In the design of effective solvers for differential equations varying the step size is crucial for their performance. Here, we implement a variable step size code for the BDF and Rosenbrock methods. As we can see in Figures 5.6, 5.9 and 5.10 their behavior is quite satisfactory, the step sizes are getting bigger when the solution is more smooth (5.6(c),(d) and 5.10(c),(d)) and the error vs. step size tend to be constant. The latter is more evident for Example 3 (5.9(c), (d), (e), (f)) and Example 4 (5.10(e) and (f)). We were particularly interested in the behavior of the variable step size Rosenbrock solver because it is cheaper to compute than a variable step size BDF solver of order two, mainly for being of one-step type. Therefore, we allowed bigger step sizes for this method. However, we can see that our variable step size solvers are sensitive to initial transients

and therefore require rather small step sizes to start up the integrator. We can clearly visualize this phenomenon in Figures 5.10(c) and (d). There, every time that the function (5.4) goes through a discontinuity point the step size is drastically reduced. In fact, the sensitivity to initial transients is a quite popular phenomenon among variable step size solvers. Hence, a priori described step sizes seem to be more practical, and cheap to compute, than variable step sizes especially for large-scale applications where we have to be concern more about computational cost and memory requirements.
If a variable step size solver has to be applied, then the Rosenbrock method of order two with variable step size is a reasonable option for the autonomous case. Note that for the non-autonomous case, the computational cost of applying the Rosenbrock methods increases considerably due to the approximation of the derivative involved. Therefore, the BDF methods are the better option there.

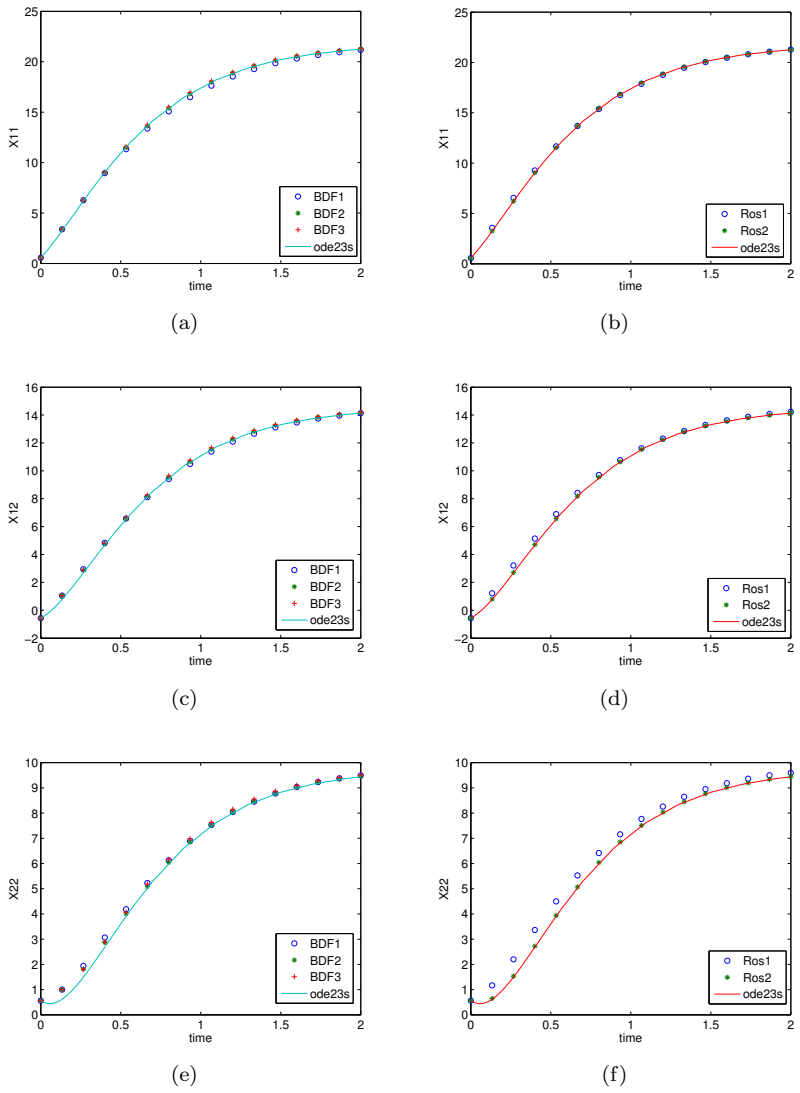


Figure 5.2: Example 1, comparison between `ode23s`, interval of integration [0,2] (a) approximate solution component X11 by the BDF methods, (b) and by the Rosenbrock methods, (c) approximate solution component X12 by the BDF methods, (d) and by the Rosenbrock methods, (e) approximate solution component X22 by the BDF methods, (f) and by the Rosenbrock methods

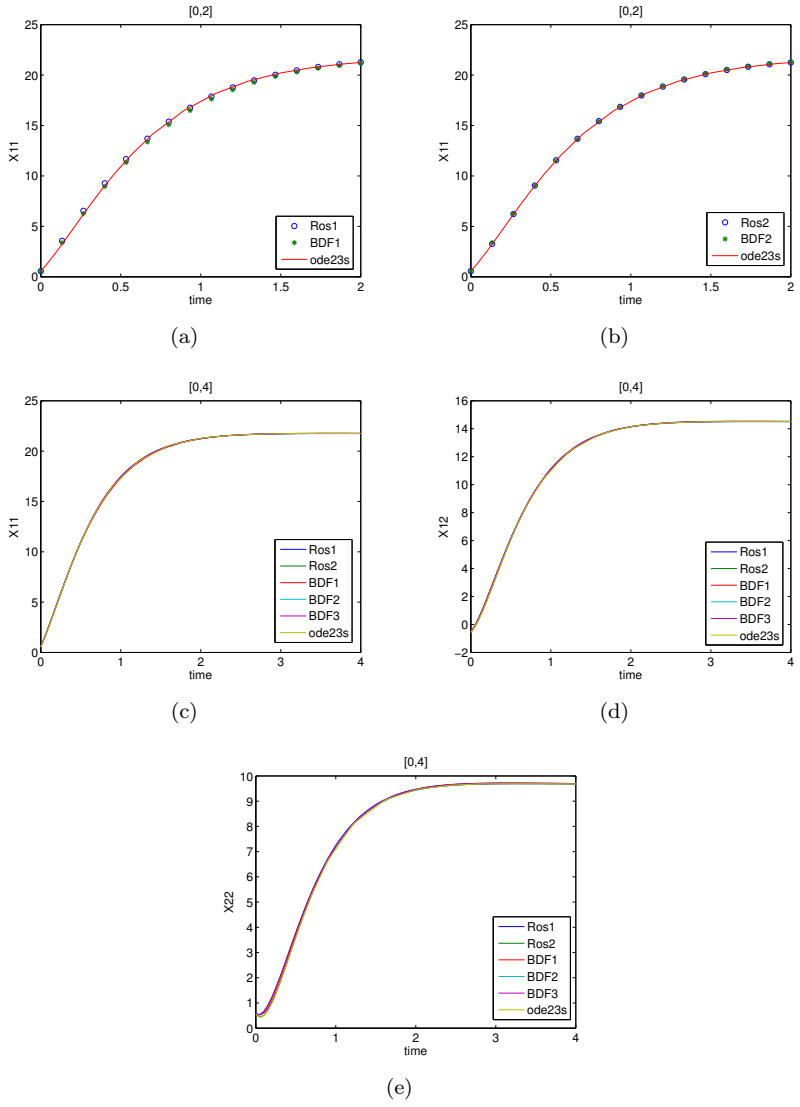


Figure 5.3: Example 1 (a) BDF1 vs linearly implict Euler method (Ros1), (b) BDF2 vs Rosenbrock method of order two (Ros2), (c) convergence to the solution of the associated ARE, component X11, (d) component X12, (e) and component X22.

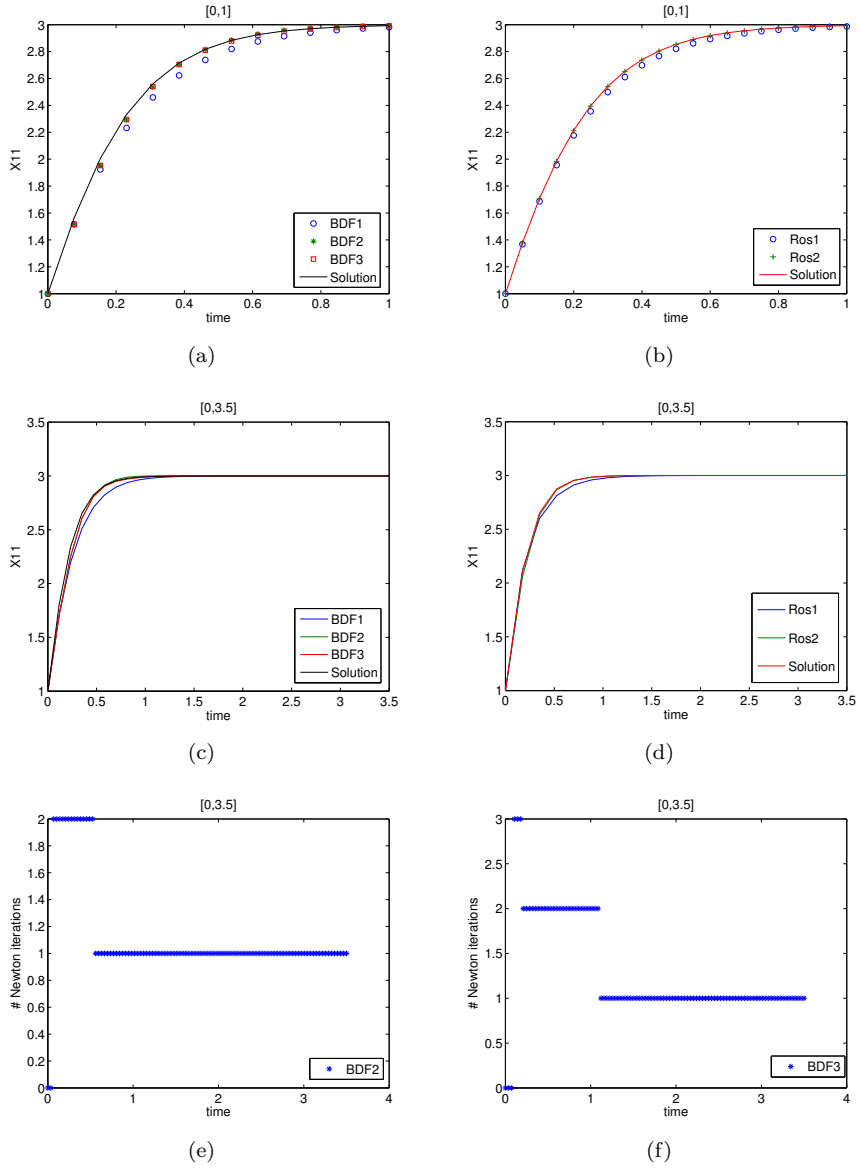


Figure 5.4: Example 2 (a) exact solution component X11 and approximated by the BDF methods, (b) and by the Rosenbrock methods, (c) convergence to the solution of the associated ARE, component X11, by the BDF methods, (d) and by the Rosenbrock methods, (e) number of Newton iterations per step for BDF2, (f) number of Newton iterations per step for BDF3.

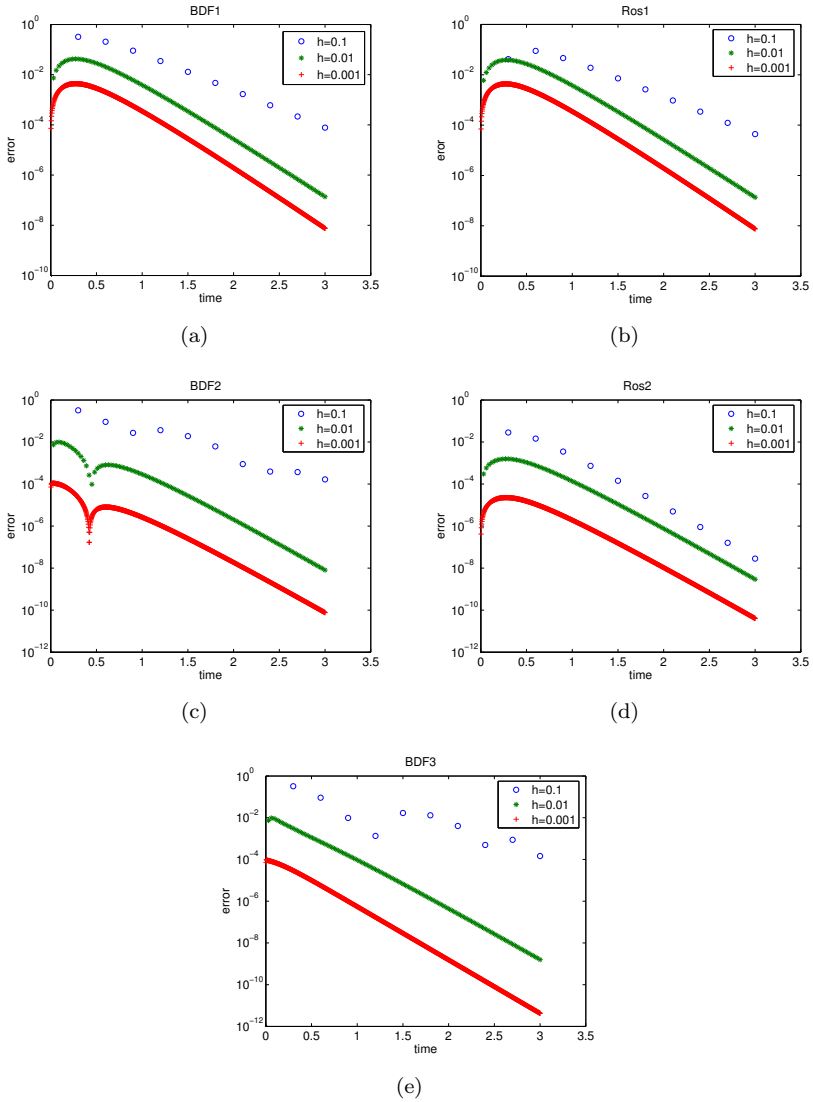


Figure 5.5: Example 2, interval of integration $[0, 3]$ (a) error per step using fixed step sizes $h = 0.1$, $h = 0.01$, and $h = 0.001$ by the BDF1, (b) by linearly implicit Euler method (Ros1), (c) by the BDF2, (d) by the Rosenbrock method of order two (Ros2), (e) and by the BDF3.

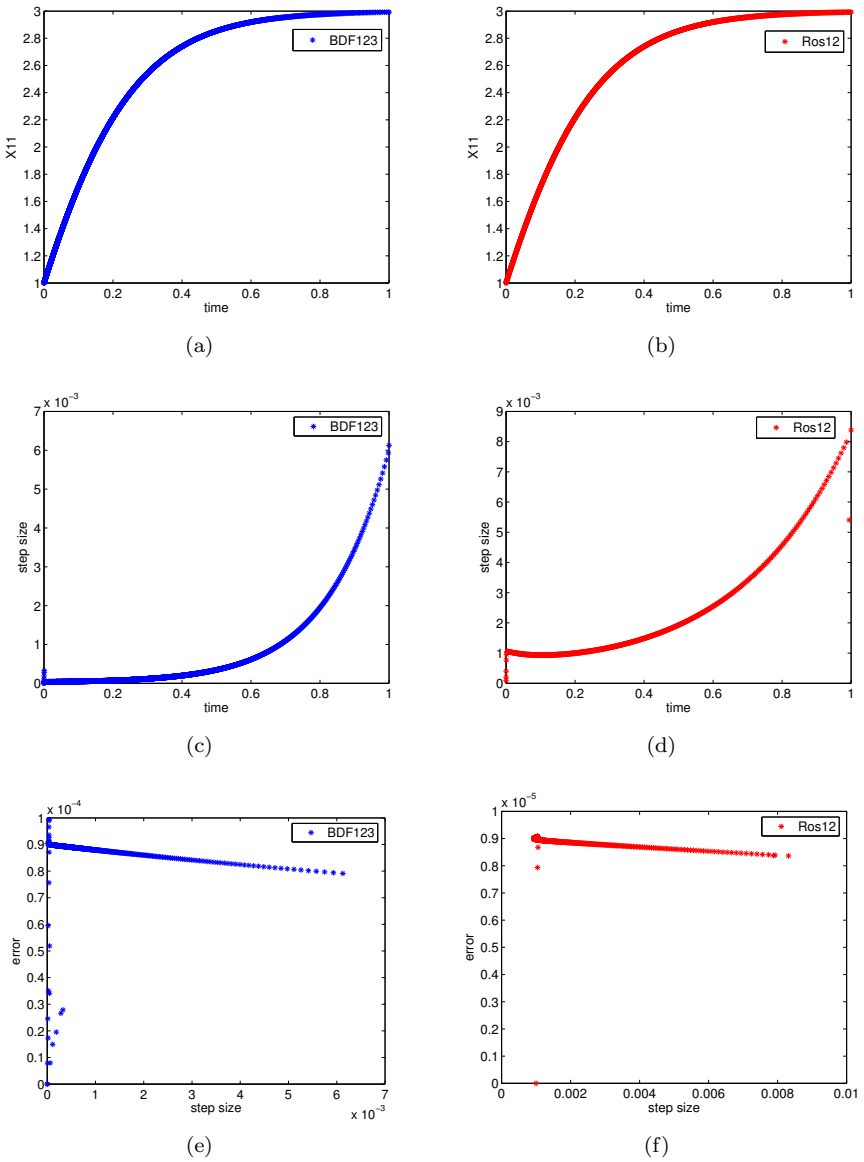


Figure 5.6: Example 2, interval of integration $[0, 1]$, $Tol = 1e{-}4$ (a) approximate solution component X11 by the variable step size and order BDF method up to order 3 (BDF123), (b) and by the variable step size Rosenbrock method of order two (Ros12), (c) step sizes over time for BDF123, (d) and step sizes over time for ROS12, (e) error vs. step size BDF123, (f) and error vs. step size ROS12.

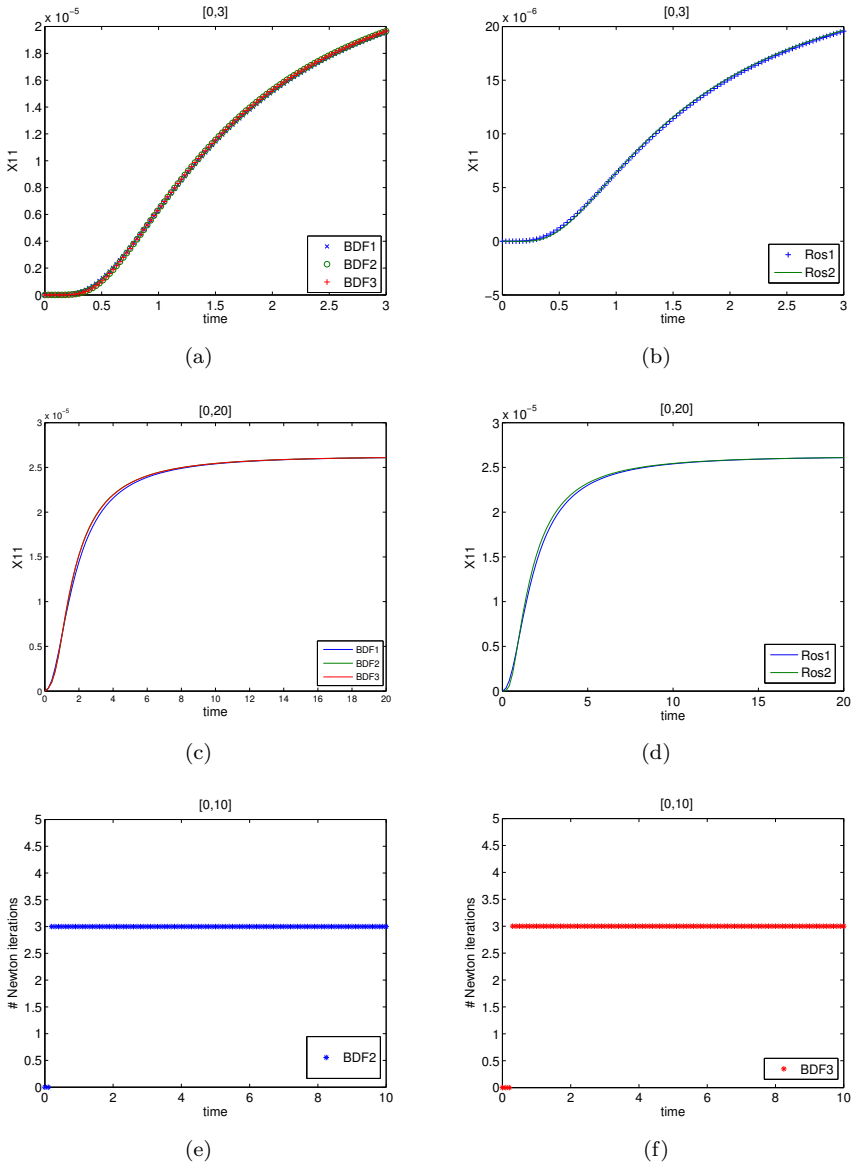


Figure 5.7: Example 3, Test 1 (a) the solution component X11 approximated by the BDF methods, (b) and by the Rosenbrock methods, (c) convergence to the solution of the associated ARE by the BDF methods, (d) and by the Rosenbrock methods, (e) number of Newton iterations per step for BDF2, (f) number of Newton iterations per step for BDF3.

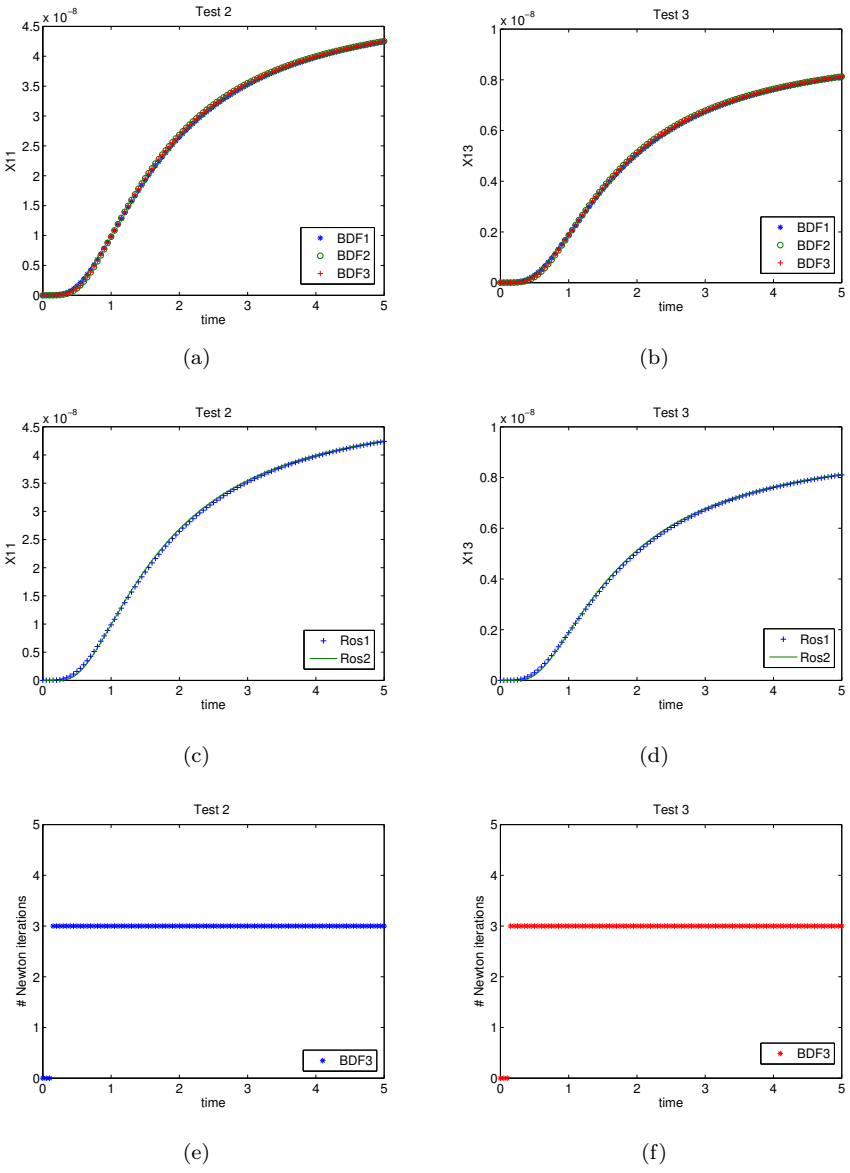


Figure 5.8: Example 3, interval of integration [0,5] (a) the solution component X11 approximated by the BDF methods, Test 2, (b) and approximated solution component X13, Test 3, (c) approximated solution component X11 by the Rosenbrock methods, Test 2, (d) and approximated solution component X13, Test 3, (e) number of Newton iterations per step for BDF3, Test2, (f) and number of Newton iterations per step for BDF3, Test 3.

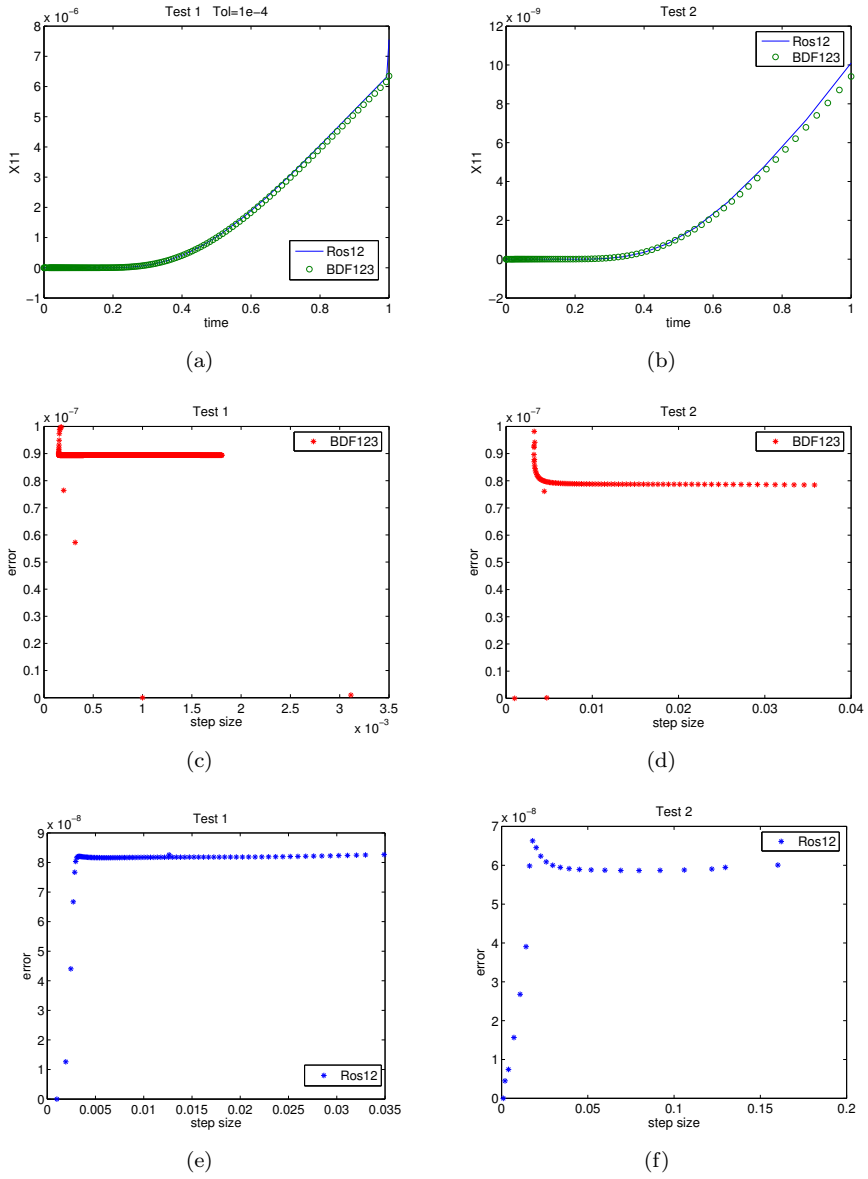


Figure 5.9: Example 3, interval of integration $[0, 1]$, $Tol = 1e{-}7$ (a) approximate solution component X11 by the variable step size and order BDF method up to order 3 (BDF123) and the Rosenbrock method of order two (Ros12) for Test 1, (b) and for Test 2, (c) error vs. step size BDF123 for Test 1, (d) and for Test 2, (e) error vs. step size Ros12 for Test 1, (f) and for Test 2.

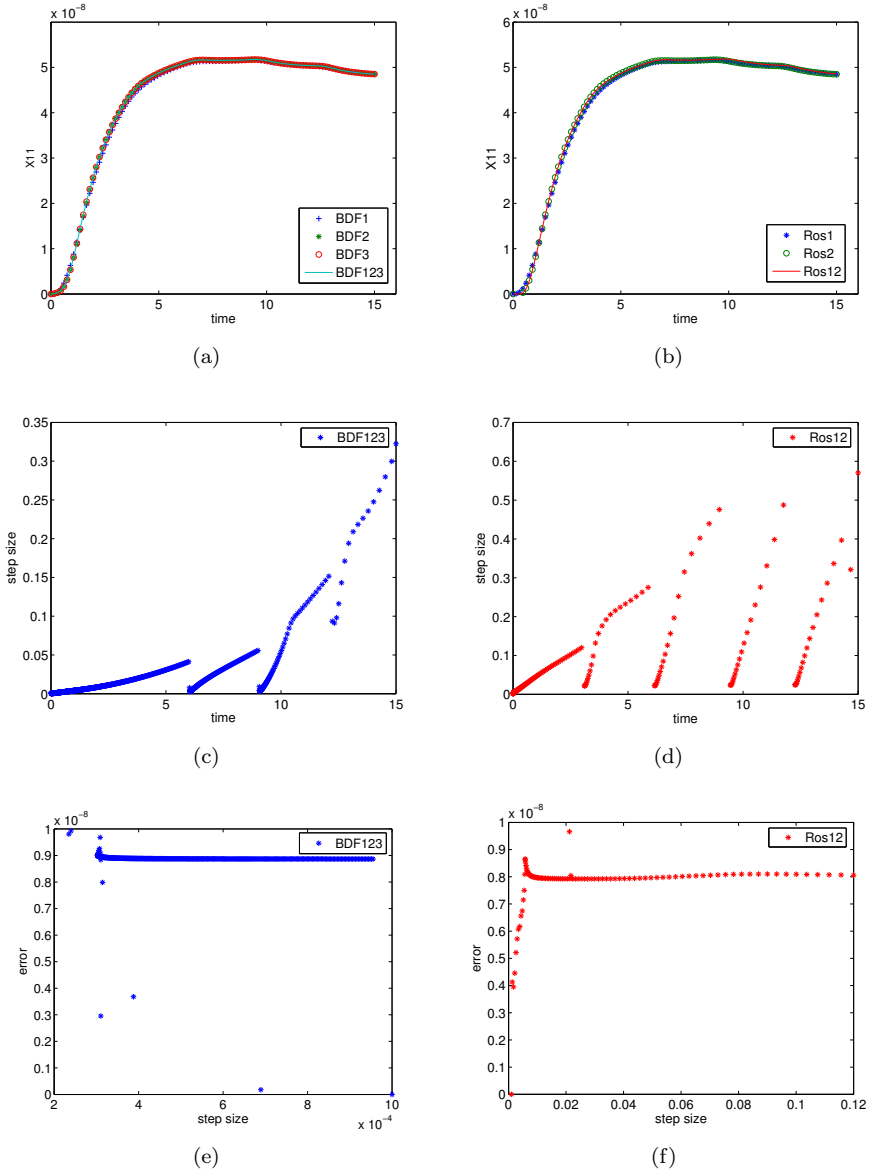


Figure 5.10: Example 4, interval of integration $[0, 15]$, $Tol = 1e-8$ (a) approximate solution component X11 by the BDF methods, (b) and by the Rosenbrock methods, (c) step sizes for the variable step size and order BDF methods up to order 3 (BDF123), (d) and for the Rosenbrock method of order two (Ros12), (e) error vs step size BDF123, (f) and Ros12.

CHAPTER

SIX

Application of DRE solvers to control problems

In this chapter we present the application of the DRE solvers discussed in this thesis to control problems. First of all, in Section 6.1, we briefly state the finite-dimensional linear-quadratic control problem and show some numerical experiments for the heat equation. Particularly, we consider the linearized version of the optimal cooling of steel profiles problem. Then, in Section 6.2, we consider the nonlinear case and summarize the idea of receding horizon techniques and its usage in a model predictive control scheme. Finally, in Section 6.3 the LQG approach for a linearization around a reference trajectory is shown as well as a numerical experiment for the Burgers equation.

6.1 The LQR problem

We consider the LQR problem:

Minimize:

$$\begin{aligned} J(x_0, u) &:= \int_0^{T_f} \langle x, Qx\rangle + \langle u, Ru\rangle dt \\ &\quad + \langle x_{T_f}, G x_{T_f}\rangle \end{aligned} \tag{6.1}$$

with respect to

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \quad t > 0, \quad x(0) = x_0. \\ y(t) &= Cx(t) \qquad\qquad t \geq 0. \end{aligned}$$

If $Q \geq 0$, $R > 0$ then, by Theorem 2.2.6 the optimal control for (6.1) is given in feedback form by,

$$u_*(t) = -K(t)^T x(t),$$

where $K(t)$ is the feedback matrix-valued function defined as,

$$K(t) = -X_*(t) B R^{-1},$$

and $X_*(t)$ is the unique nonnegative self-adjoint solution of the differential Riccati equation:

$$\begin{aligned} &\dot{X}(t) = -(C^T Q C + A^T X(t) + X(t)A - X(t)BR^{-1}B^T X(t)), \\ &X(T_f) = G. \end{aligned} \tag{6.2}$$

In Chapter 4, we proposed efficient methods to compute (6.2) based on a low rank version of the ADI iteration. The low rank factors delivered by these methods in general contain more columns than the feedback matrix $K(t)$. The computation of feedback matrices directly (i.e., without explicitly computing the low rank factors approximating the solution of (6.2)) unfortunately is not possible for our methods. This is due to the fact that, in the right hand side of the Lyapunov equation involved in the solution of the DRE, using the BDF or the Rosenbrock methods, the previous computed step(s) appear(s) explicitly, i.e. X_k, X_{k-1}, ... Therefore the low rank factor of it(them) is(are) needed to compute the next step.
First of all, notice that due to the symmetry and definiteness assumptions, the matrices Q and R can be factorized as

$$Q = \tilde{Q}\tilde{Q}^T, \qquad R = \tilde{R}^T\tilde{R}, \tag{6.3}$$

where $\tilde{Q} \in \mathbb{R}^{p\times q}$ $(q \leq p)$ and $\tilde{R} \in \mathbb{R}^{m\times m}$. If we denote $\tilde{C} = \tilde{Q}^T C$ and $\tilde{B} = BR^{-1}\tilde{R}$, then (6.2) can be expressed in the form

$$\begin{aligned} &\dot{X}(t) = -\tilde{C}^T\tilde{C} - A^T X(t) - X(t)A + X(t)\tilde{B}\tilde{B}^T X(t), \\ &X(T_f) = G. \end{aligned}$$

By Remark 2.2.7, we can solve (forward) in time the DRE

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

and afterwards recover the solution of (6.2). The latter equation has the form of (4.1), which was the one considered in Chapter 4. Hence, we are able to directly apply our methods to solve (6.2).

6.1.1 Numerical experiments

We now present numerical experiments for the linear-quadratic regulator problem. The Lyapunov equation involved in the solution of the DRE by the BDF, or the Rosenbrock, methods is solved using the LyaPack software package [94] using the ADI parameter selection method proposed in Section 4.4. In our implementation we keep the concept of **user-supplied functions** introduced in LyaPack meanwhile the matrix operations with A (multiplications, solution of systems of (shifted) linear equations) are realized implicitly. The data related to matrix A is stored in global variables making the routines efficient for both memory and computation, see [94] for details. We performed preprocessing (and

Test	n	n_0	Q	R	G
1	400	20	I	I	0
2	625	25	10I	I	0
3	900	30	10I	I	0

Table 6.1: Parameters for FDM semi-discretized heat equation.

post-processing as well) of the dynamical system reordering the nonzero pattern of A for bandwidth reduction. The efficiency of our methods strongly depends on the way how these operations are computed, e.g., if A is sparse and symmetric, linear systems are solved by sparse Cholesky factorization. For memory efficient storage the feedback matrix is stored in every step instead of the low rank factor of the approximate solution.
We solve the DRE as well as the closed-loop system using fixed step size. The latter was computed by the implicit Euler method using the Sherman-Morrison-Woodbury formula to efficiently solve the linear system involved.

FDM semi-discretized heat equation. We consider the finite difference semi-discretized heat equation on the unit square $(0,1) \times (0,1)$.

$$\frac{\partial \mathbf{x}}{\partial t} - \Delta \mathbf{x} = \mathbf{f}(\xi)\mathbf{u}(t). \tag{6.4}$$

In Figure 4.3(a) we plot the sparsity pattern of A. The data is generated by the routines `fdm_2d_matrix` and `fdm_2d_vector` from the examples of the LyaPack software package. Details on the generation of test problems can be found in the documentation of these routines (comments and MATLAB help).
The problem parameters chosen here are shown in Table 6.1, there n is the problem dimension, n_0 is the number of grid points in either space direction, and Q, R, G are the operators from the LQR problem.
The convergence history, after fifty iterations, for the Lyapunov equation (left) and ARE (right) involved in the solution of the DRE using the BDF method of order one are shown in Figure 6.1 for the different tests. As a test example we use here the following stopping criterion: stagnation of the normalized residual norm. Note that the computation of the normalized residual norm is expensive and can even exceed the computational cost of the iteration itself, [94]. Hence, we avoid this stopping criterion in the following.

Optimal cooling of steel profiles. Let us consider the problem of optimal cooling of steel profiles, [20, 23, 24, 49, 103, 112]. This problem arises in a rolling mill when different steps in the production process require different temperatures of the raw material. To achieve a high production rate, economical interests suggest to reduce the temperature as fast as possible to the required level before entering the next production phase. At the same time, the cooling process, which is realized by spraying cooling fluids on the surface, has to be

controlled so that material properties, such as durability or porosity, achieve given quality standards. Large gradients in the temperature distributions of the steel profile may lead to unwanted deformations, brittleness, loss of rigidity, and other undesirable material properties. It is therefore the engineers goal to have a preferably even temperature distribution.

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 6.2. A (linearized) version of the model has the form

$$\begin{array}{rcll} c\varrho x_t(\xi,t) & = & \lambda\Delta x(\xi,t) & \text{in } \Omega\times(0,T), \\ -\lambda\partial_\nu x(\xi,t) & = & g_i(t,x,u) & \text{on } \Gamma_i \text{ where } i=0,\ldots,7, \\ x(\xi,0) & = & x_0(\xi) & \text{in } \Omega, \end{array} \tag{6.5}$$

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.
After discretization in space we get a model of the form

$$\begin{aligned} M\dot{\tilde{x}}(t) &= N\tilde{x}(t)+\tilde{B}u(t), \\ y(t) &= \tilde{C}\tilde{x}(t), \end{aligned}$$

where M, $N\in\mathbb{R}^{n\times n}$, M is invertible and $M^{-1}N$ is stable. The inverse matrix of M is computed by LU factorization, i.e., $M=M_LM_U$. Then the standard form of the system is recovered by

$$A=M_L^{-1}NM_U^{-1},\quad B=M_L^{-1}\tilde{B},\quad C=\tilde{C}M_U^{-1}.$$

We point out that matrix A is not computed explicitly and the operations related to the matrix are done implicitly. The initial condition and computational mesh of the numerical test is shown in Figure 6.3.
We applied the BDF method of order one with fixed step size for $n=1357$. For the refined mesh, case $n=5177$, the linearly implicit Euler method (Rosenbrock method of order one) was applied. The problem parameters chosen can be found in Table 6.2. 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 six control parameters over time in Figure 6.4 for $n=1357$, and for $n=5177$ in Figure 6.5. They converge to zero because $G=0$ and therefore the final feedback matrix as well as the control are equal zero. In Table 6.3 the cost functional values are shown. The values from the finite-time horizon case (DRE) are smaller than for the infinite-time horizon case (ARE).

Test	n	Q	R	G	T_f	h
1	1357	I	I	0	20	0.01
2	5177	I	I	0	20	0.01

Table 6.2: Parameters for cooling of steel profiles problem.

n	DRE	ARE
1357	2.1601 e+06	5.0823 e+07
5177	1.9834 e+06	4.0613 e+07

Table 6.3: Cost functional values for finite-time horizon (DRE) and infinite-time horizon (ARE).

6.2 Usage of LQR design in MPC scheme

We briefly summarize now the usage in a MPC scheme similar to [17, 68].
Let us consider the optimal control problem

$$\begin{aligned} &\text{Minimize:} \\ &\qquad \min \textstyle\int_0^{T_f} f^0(y(t),u(t))dt \\ &\text{with respect to} \\ &\qquad \dot{x}(t) = f(x(t)) + Bu(t),\ t>0,\ x(0)=x_0, \\ &\qquad y(t) = Cx(t) \qquad\qquad t \geq 0. \end{aligned} \tag{6.6}$$

where $T_f \in [0,\infty[$, $x_0 \in \mathbb{R}^n$, and f is a nonlinear function. We assume here that the state space is finite-dimensional to avoid difficulties associated to infinite-dimensional control systems, see [67]. The solution of (6.6) can be found solving the system resulting from the application of the minimum principle or constructing the feedback solution based on Bellman's dynamic programming. In both cases, the numerical solution represents a computational challenge.
An alternative is to apply receding horizon techniques, based on model predictive control which we briefly explain in the following.
Let $0 = T_0 < T_1 \cdots < T_f$ describe a grid on $[0,T_f]$ and let $T \geq \max\{T_{i+1} - T_i : i = 0,\dots\}$. Based on the MPC approach (see, e.g, [4, 51]) we have to solve the successive finite horizon optimal control problems on $[T_i, T_i + T]$,

$$\begin{aligned} &\text{Minimize:} \\ &\qquad \min \textstyle\int_{T_i}^{T_i+T} f^0(y(t),u(t))dt + \tilde{G}(x(T_i+T)) \\ &\text{with respect to} \\ &\qquad \dot{x}(t) = f(x(t)) + Bu(t),\ t>0, \end{aligned} \tag{6.7}$$

where $x(T_i) = x_i^*(T_i)$ for $i \geq 1$ and $x(0) = x_0$ for $i = 0$. Here x_i^* is the solution on the previous time frame $[T_{i-1}, T_{i-1}+T]$. The cost functional contains a terminal cost $\tilde{G}$ to penalize the states at the end of the finite horizon, if $\tilde{G}$ is chosen as a

control Liapunov function, then the asymptotic stability and the performance estimate of the receding horizon synthesis are established in [66], for the case in which the state space is finite-dimensional and in [67], for infinite-dimensional state spaces. Another possibility to guarantee stability of the closed-loop system is to add additional constraints to the problem, for example $x(T_i+T) \in \Omega$. This constraints force the states at the end of the prediction horizon to be in some neighborhood Ω (terminal region) of the target.
The solution on $[0, T_f]$ is obtained by concatenation of the solutions on $[T_i, T_{i+1}]$ for $i = 0, \ldots$. The optimal control for the problem on $[T_i, T_{i+1}]$ is computed via an linear-quadratic Gaussian (LQG) approach. If $x(T_i)$ is observed, this technique is a feedback method since the control on $[T_i, T_{i+1}]$ is determined as a function of the state $x^*(T_i)$. We point out that it is also possible to apply LQR instead of the LQG approach to compute the optimal control for the problem on $[T_i, T_{i+1}]$, however in general small noises will lead to large deviations. This results in useless solutions or in large jumps of the controller.

6.3 Linear-quadratic Gaussian control desing

The linear-quadratic Gaussian (LQG) approach is an extension of the LQR approach which allows noise and includes observer, see for instance [84]. It arises in a large number of areas of engineering, aerospace and economics, as well as in situations in which the initially nonlinear dynamics are linearized around a reference trajectory. In the following we review the latter.
Let us consider a nonlinear stochastic control system

$$\dot{x}(t) = f(x(t)) + Bu(t) + Fv(t),\ x(0) = x_0, \tag{6.8}$$

where $v(t)$ is an unknown Gaussian disturbance process.
The observation process

$$y(t) = Cx(t) + w(t) \tag{6.9}$$

provides partial observations of the state $x(t)$, where $w(t)$ is a measurement noise process which will also be assumed to be Gaussian.
Let $x^*(t)$ be a reference trajectory and $u^*(t)$ the associated control. We define the errors

$$\delta x(t) = x(t) - x^*(t), \quad \delta u(t) = u(t) - u^*(t),$$

and consider

$$\begin{aligned} \frac{d}{dt}(x^*(t) + \delta x(t)) &= f(x^*(t) + \delta x(t)) + B(u^*(t) + \delta u(t)) + Fv(t), \\ x(0) &= x_0 + \eta_0. \end{aligned}$$

If we expand $f(x^*(t) + \delta x(t))$ up to first order, we can replace it by $f(x^*(t)) + f'(x^*(t))\delta x(t)$. Since x^* satisfies the equation (6.8) we get

$$\frac{d}{dt}(x(t) - x^*(t)) \approx A(t)(x(t) - x^*(t)) + B(u(t) - u^*(t)) + Fv(t),$$

where $A(t) = A(x^*(t)) = f'(x^*(t))$.
Let

$$z(t) = x(t) - x^*(t), \quad \tilde{u}(t) = u(t) - u^*(t),$$

then, we obtain the time-varying system

$$\dot{z}(t) = A(t)z(t) + B\tilde{u}(t) + Fv(t), \quad z(0) = \eta_0.$$

Let $Q \in \mathbb{R}^{n\times n}$ denote a positive definite matrix and consider the tracking problem for the pair (x^*, u^*)

Minimize:

$$J(z_0, \tilde{u}) := \frac{1}{2}\int_0^{T_f} z(t)^T C^T Q C z(t) + \tilde{u}(t) R \tilde{u}(t) dt + z(T_f)^T G z(T_f)$$

with respect to

$$\begin{aligned} \dot{z}(t) &= A(t)z(t) + B\tilde{u}(t) + Fv(t), \quad z(0) = \eta_0, \\ y(t) &= Cx(t) + w(t), \qquad\qquad\quad t \geq 0. \end{aligned}$$

For the feedback law we use an estimated state of the process which is based on the measured output $\tilde{y}$, i.e, we use

$$\tilde{u}(t) = -K(t)\hat{z}(t)$$

where $\hat{z}(t)$ denotes the estimated state of the system.
If we use a Kalman filter, see for instance [34], then the estimated state $\hat{z}(t)$ is given by

$$\dot{\hat{z}}(t) = A(t)\hat{z}(t) + B\tilde{u}(t) + L(t)(y(t) - C\hat{x}(t)).$$

The feedback law can be represented as

$$u(t) = u_*(t) + K(t)^T(\hat{x}(t) - x^*(t)),$$

where $K(t)$ is the feedback matrix defined as

$$K(t) = -X_*(t)BR^{-1},$$

and $X_*(t)$ is the unique nonnegative self-adjoint solution of the differential Riccati equation:

$$\dot{X}(t) = -(C^T Q C + A(t)^T X(t) + X(t)A(t) - X(t)BR^{-1}B^T X(t)). \tag{6.10}$$

Theorem 6.3.1 *Let the following conditions hold, see the Appendix A:*

(i) (A,B,C) is controllable and observable.

(ii) v and w are white noise, zero-mean stochastic processes, that is for all t,s

$$\begin{aligned} \mathbb{E}[v(t)] &= 0, \\ \mathbb{E}[w(t)] &= 0, \\ \mathbb{E}[v(t)v^T(s)] &= V\delta(t-s), \\ \mathbb{E}[w(t)w^T(s)] &= W\delta(t-s), \end{aligned}$$

where $V := \mathrm{cov}(v(t))$ is symmetric, positive semi-definite, $W := \mathrm{cov}(w(t))$ is symmetric, positive definite and δ is the Dirac function. Furthermore, it is assumed that V and W are time-independent.

(iii) v and w are uncorrelated, that is $\mathbb{E}[v(t)w^T(s)] = 0$, for all t, s.

Then the best estimate $\hat{x}(t)$ of $x(t)$ can be generated by the Kalman filter

$$\dot{\hat{x}}(t) = A(t)(\hat{x}(t) - x^*(t)) + f(x^*(t)) + Bu(t) + L(t)(C(x(t) - \hat{x}(t)) + w(t)),$$

where the filter gain matrix $L(t)$ is given by

$$L(t) = \Sigma^*(t)C^T W^{-1}$$

and $\Sigma^*(t)$ is the symmetric solution of the filter differential Riccati equation (FDRE)

$$\dot{\Sigma}(t) = F^T V F + A(t)\Sigma(t) + \Sigma(t)A(t)^T - \Sigma(t)C^T W^{-1} C\Sigma(t). \tag{6.11}$$

Proof. The proof of this theorem can be found for instance in [84].

Algorithm 6.3.1 sketches the LQG approach.

Remark 6.3.2 *The LQG design (approach) for a linearization around an operating point will lead to an algorithm similar to Algorithm 6.3.1 in which the AREs:*

$$0 = C^T Q C + A^T X + XA - XBR^{-1}B^T X, \tag{6.12}$$
$$0 = FVF^T + A\Sigma + \Sigma A^T - \Sigma C^T W^{-1} C\Sigma, \tag{6.13}$$

have to be solve instead of the DREs in step 3 (6.12) and step 6 (6.13) respectively, [84, 17, 68].

6.3.1 Numerical experiments

MPC for Burgers equation. The Burgers equation is used as a model for description of basic phenomena of flow problems like: shock waves, traffic flows, etc. Here we consider an optimal control problem of the form (6.6), subject to the Burgers equation

$$\begin{array}{rcll} x_t(t,\xi) & = & \nu x_{\xi\xi}(t,\xi) - x(t,\xi)x_\xi(t,\xi) + B(\xi)u(t) + F(\xi)v(t), & \\ x(t,0) & = & x(t,1) = 0, & t > 0, \\ x(0,\xi) & = & x_0(\xi) + \eta_0(\xi), & \xi \in]0,1[\end{array} \tag{6.14}$$

where t is the variable in time, ξ the variable in space, and ν is a viscosity parameter, and the observation process

$$y(t,\xi) = Cx(t,\xi) + w(t,\xi).$$

Algorithm 6.3.1 LQG for a linearization around the reference trajectory

Require: $A(t)$, B, C, Q, R, V, W and T.

Ensure: the optimal control $u_{opt}(t)$, in the interval $[0, T_f]$.

1: **while** $T_i \leq T_f$ **do**

2: Determine $A(t) := f'(x^*(t))$.

3: Solve the DRE

$$\dot{X}(t) = -(C^T Q C + A(t)^T X(t) + X(t)A(t) - X(t)BR^{-1}B^T X(t))$$

satisfying $X(T_i + T) = G$.

4: Let X_* be the solution of the DRE.

5: Compute the feedback matrix $K(t) = -X_*(t)BR^{-1}$

6: Solve the FDRE

$$\dot{\Sigma}(t) = FVF^T + A(t)\Sigma(t) + \Sigma(t)A(t)^T - \Sigma(t)C^T W^{-1} C\Sigma(t).$$

satisfying $\Sigma(T_i) = \Sigma_i$.

7: Let Σ_* be the solution of the FDRE.

8: Compute the filter gain matrix $L(t) = \Sigma^*(t)C^T W^{-1}$.

9: Calculate $\hat{x}(t)$ from the compensator equation

$$\begin{aligned} \dot{\hat{x}}(t) &= \dot{x}^*(t) + A(t)(\hat{x}(t) - x^*(t)) - BK^T(\hat{x}(t) - x^*(t)) \\ &\quad + L(t)(y(t) - C\hat{x}(t)), \\ \hat{x}(T_i) &= x_i^*, \end{aligned}$$

using (6.8) and (6.9) for simulating the measurements $y(t)$.

10: Determine the optimal control on $[T_i, T_i + T]$,

$$u^*_{T_i}(t) = u^*(t) + K^T(\hat{x}(t) - x^*(t)).$$

11: Add $u^*_{T_i}(t)$ to the optimal control on the whole interval

$$u_{opt}(t) = u^*_{T_i}(t), \qquad t \in [T_i, T_i + T[.$$

12: Update $T_i := T_i + T$.

13: **end while**

Test	n	Q	R	G	B	C	F	V	W	T_f	h
1	31	0.1I	0.001I	0	I	I	I	4I	0.01I	3	0.03
2	201	0.1I	0.001I	0	I	I	I	4I	0.01I	3	0.005

Table 6.4: Parameters for MPC for Burgers equation.

n	Noise in initial condition	DRE	ARE
31	0	0.0098	0.0115
	1	0.0114	0.0131
201	0	0.0080	0.097
	1	0.0128	0.0146

Table 6.5: Cost functional values with(out) noise in the initial condition.

The aim is to control the state to 0. The uncontrolled solution is plotted in Figure 6.6(a) and the reference trajectory in (b).
After discretizing (6.14) in space by using finite elements a system of the form (6.8) is obtained. The problem parameters can be found in Table 6.4. In addition we chose

$$\begin{aligned}&\mathbb{E}[v] = \mathbb{E}[w] = \mathbb{E}[\eta_0] = 0,\\ &\sigma_v = 2,\ \sigma_w = 0.1,\ \sigma_{\eta_0} = 0.3\end{aligned}$$

and the initial condition as

$$x_0(\xi) = \begin{cases} 0.3\sin(2\pi t - \pi) & \text{in }]0, \frac{1}{2}] \\ 0 & \text{in }]\frac{1}{2}, 1] \end{cases} .$$

We applied the BDF method of order one with fixed step size for solving DREs and compare our results with an LQG design approach for a linearization around an operating point, i.e, the case in which AREs are solved instead of DREs. For a discussion on LQG design approach for a linearization around an operating point we refer the reader to [17, 68].
The cost functional values are shown in Table 6.5. As for the cooling of steel profiles problem, the values using the DRE are smaller than for the ARE.
The control and the state without considering noise in the initial condition is shown in Figure 6.7 and Figure 6.8, respectively. The same pictures for a refined mesh are plotted in Figures 6.11 and 6.12. For the case in which noise in the initial condition is considered, they are plotted in Figures 6.9 and 6.10. Again, the same pictures for a refined mesh are plotted in Figures 6.13 and 6.14.
The control (the state) for the ARE and DRE look quite similar in both cases. However, a smaller cost is obtained for the case in which DREs are used.

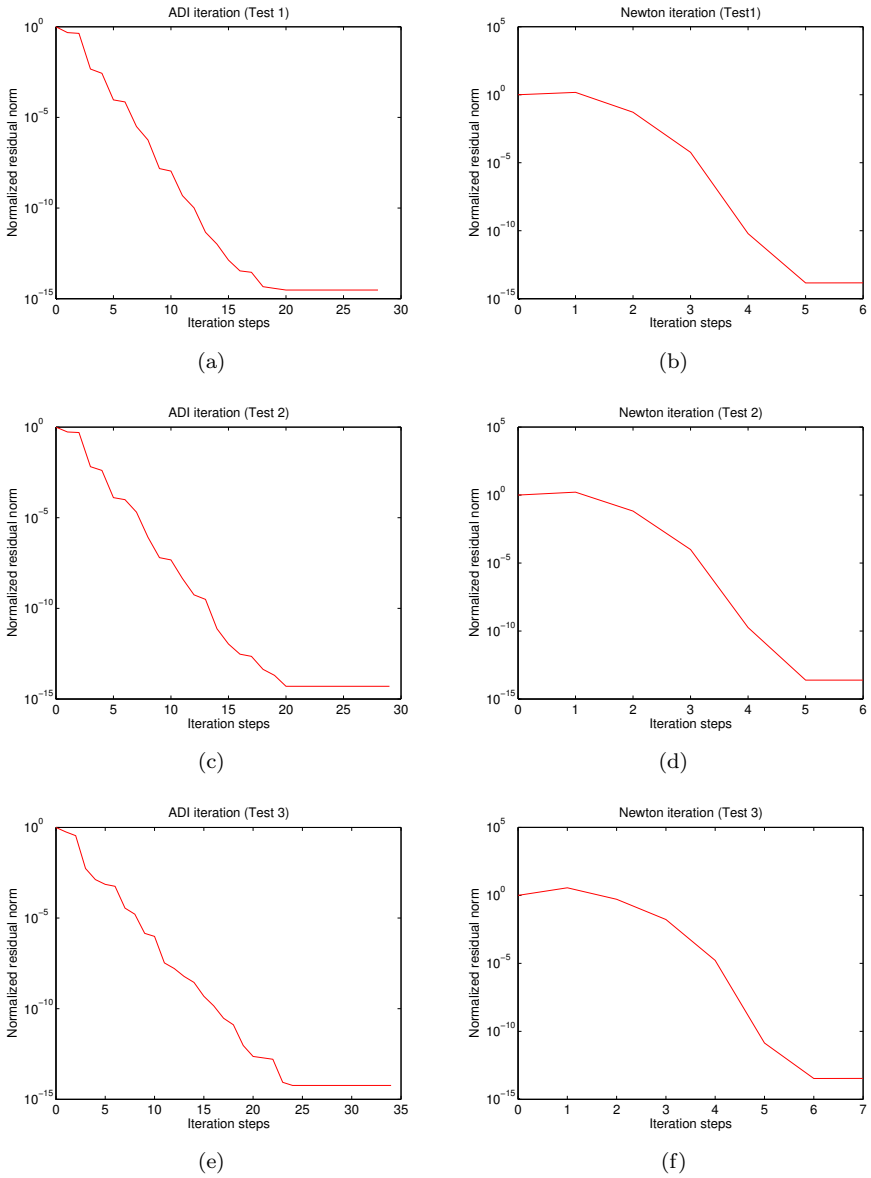


Figure 6.1: FDM semi-discretized heat equation (convergence history) (a) Low rank ADI iteration and (b) Newton iteration for Test 1, (c) low rank ADI iteration and (d) Newton iteration for Test 2, (e) low rank ADI iteration and (f) Newton iteration for Test 3.

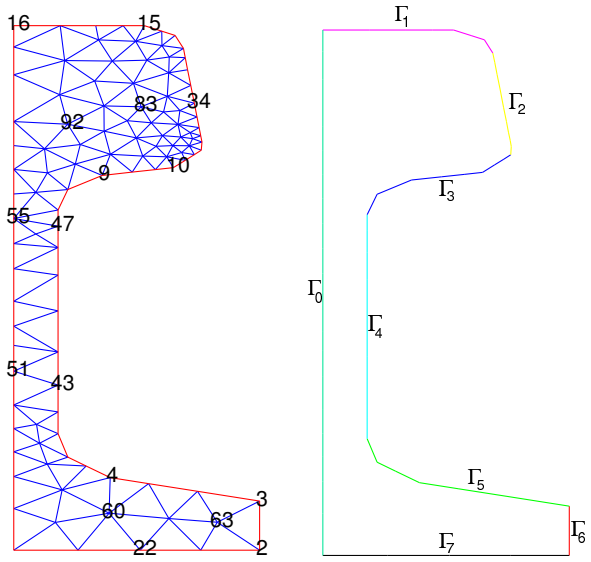


Figure 6.2: initial mesh with points of minimization (left) and partition of the boundary (right).

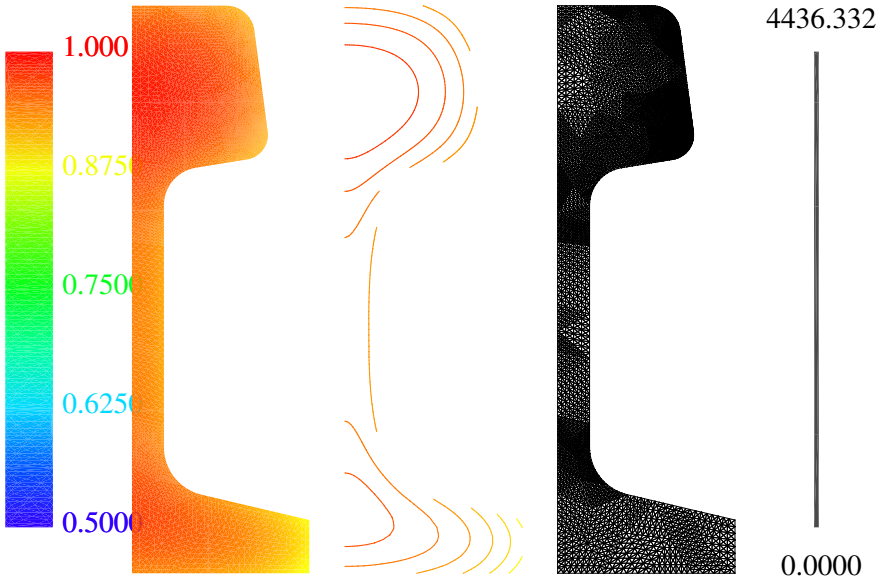


Figure 6.3: initial condition and computational mesh of the numerical test.

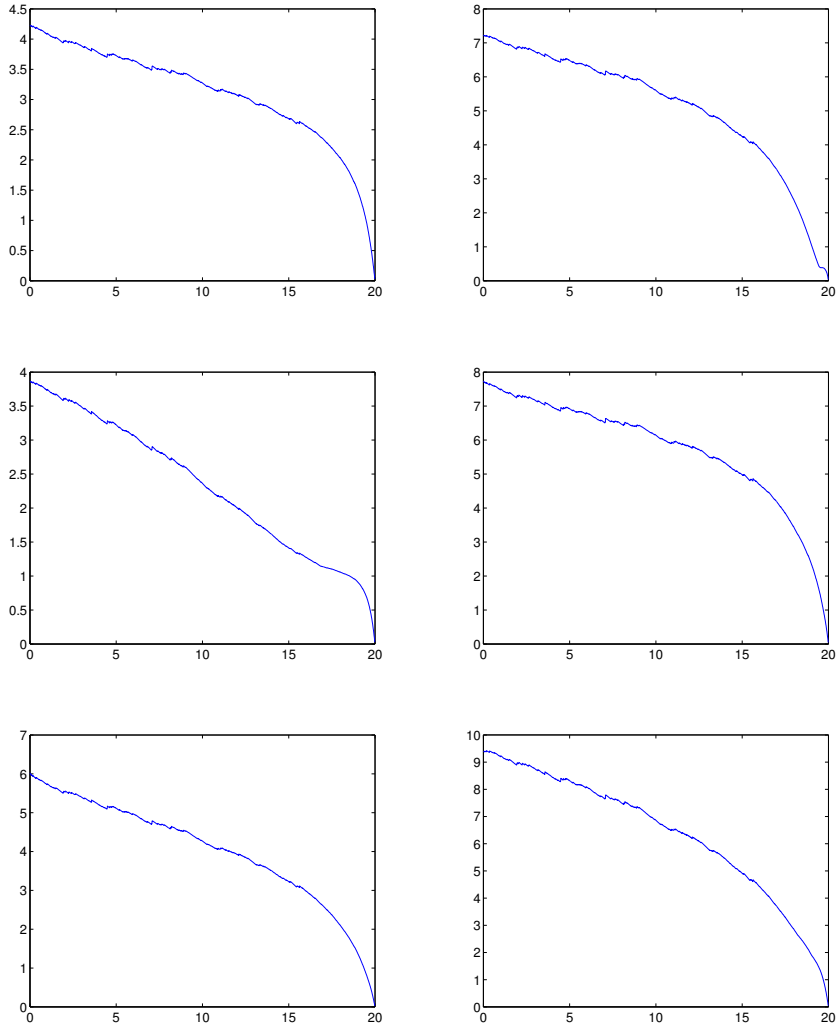


Figure 6.4: Cooling of steel profiles control parameters plotted over time for n=1357.

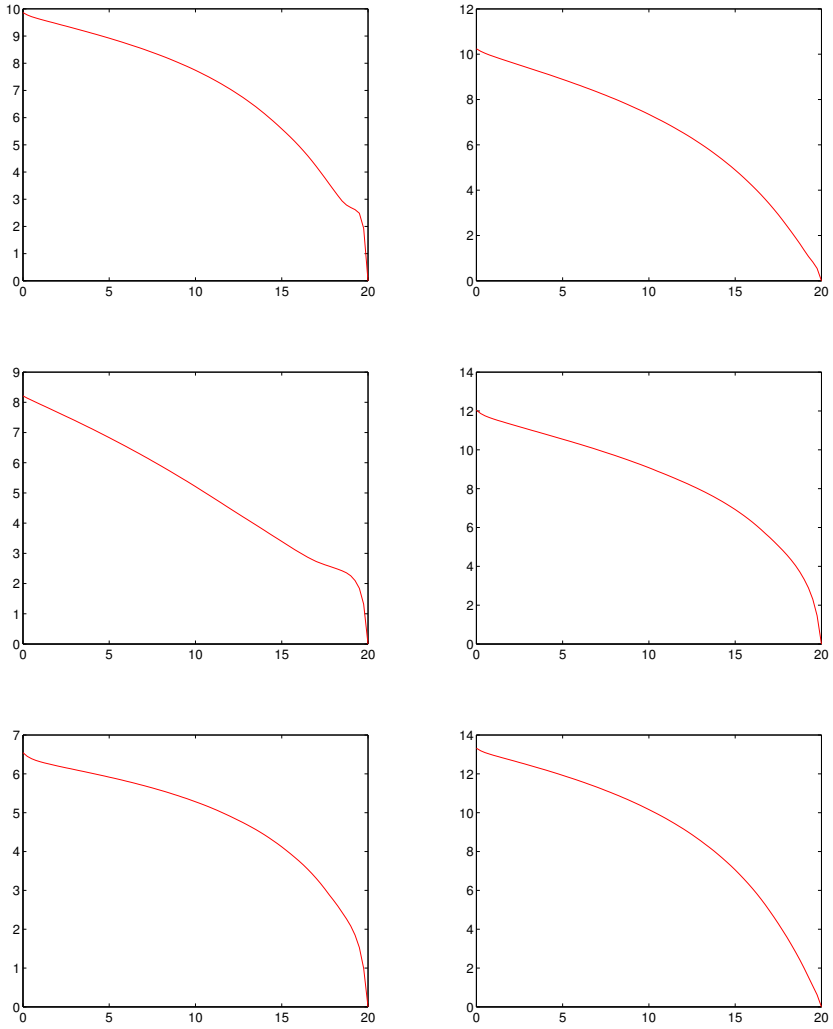
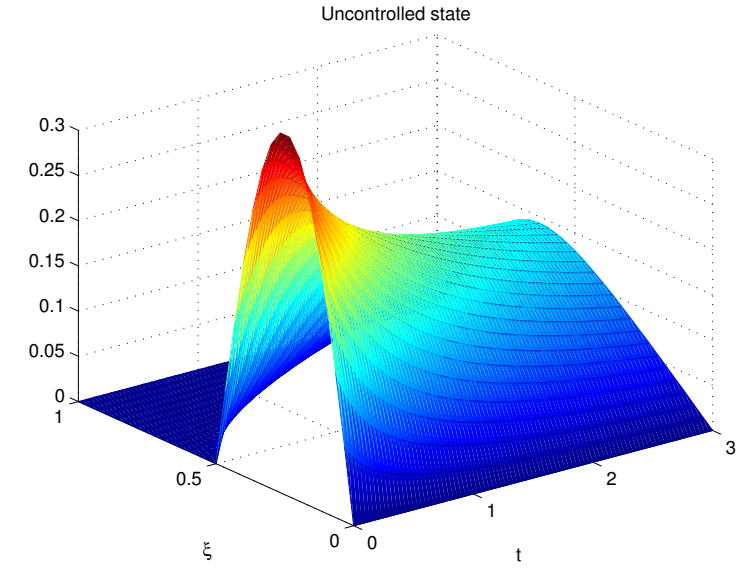
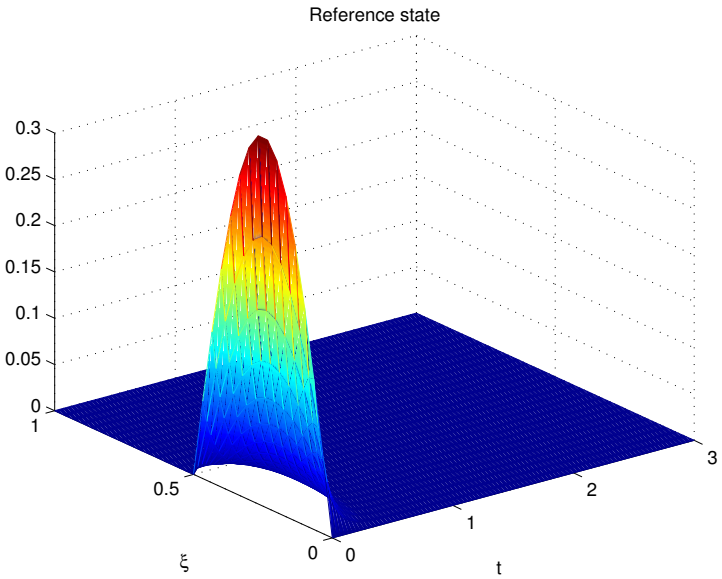


Figure 6.5: Cooling of steel profiles control parameters plotted over time for n=5177.

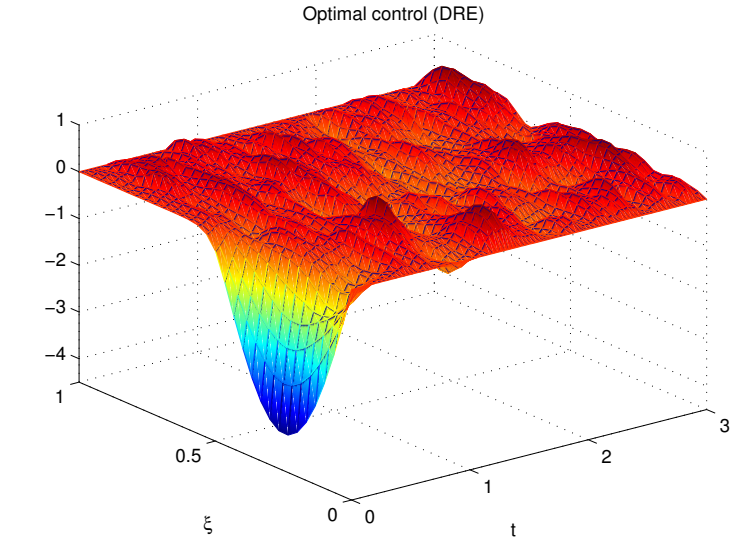


(a)

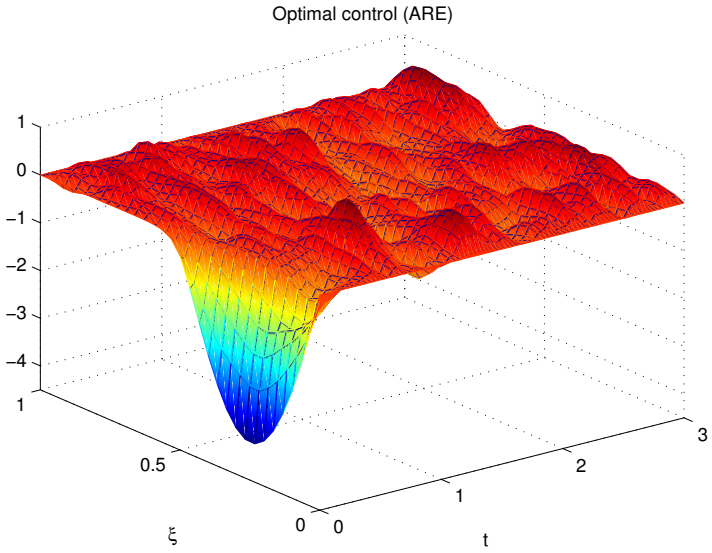


(b)

Figure 6.6: Burgers equation (a) uncontrolled solution and (b) reference state.

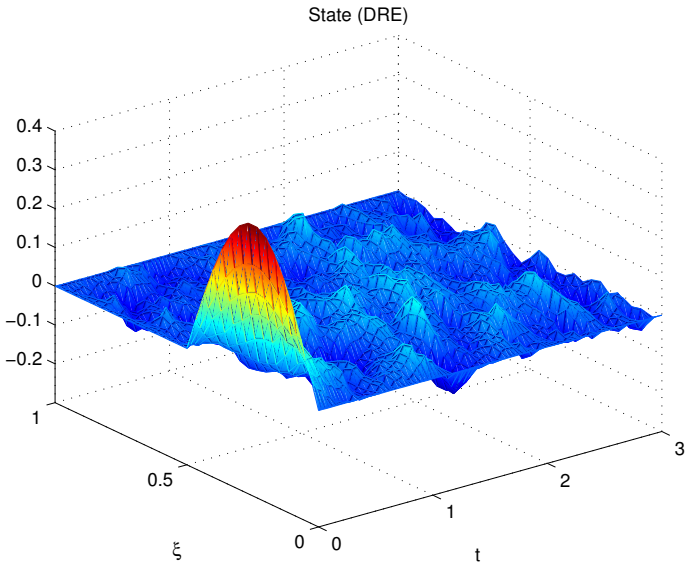


(a)

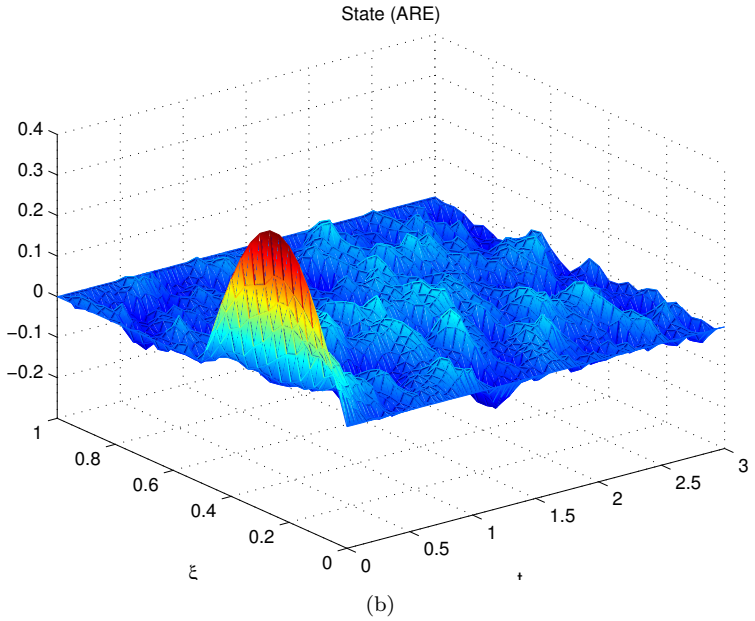


(b)

Figure 6.7: Burgers equation (a) optimal control (DRE) and (b) (ARE) for initial mesh.

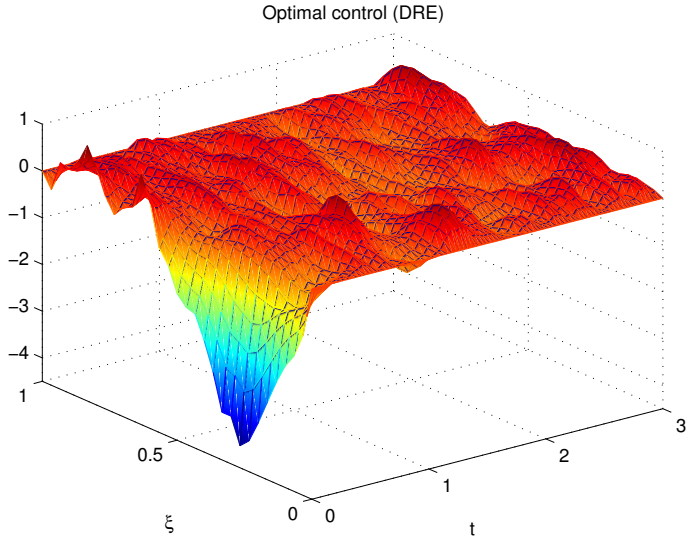


(a)

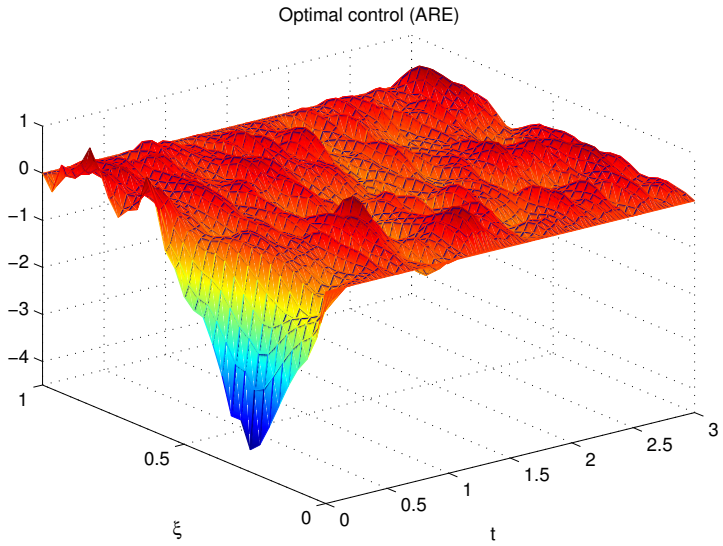


(b)

Figure 6.8: Burgers equation (a) state (DRE) and (b) (ARE) for initial mesh.

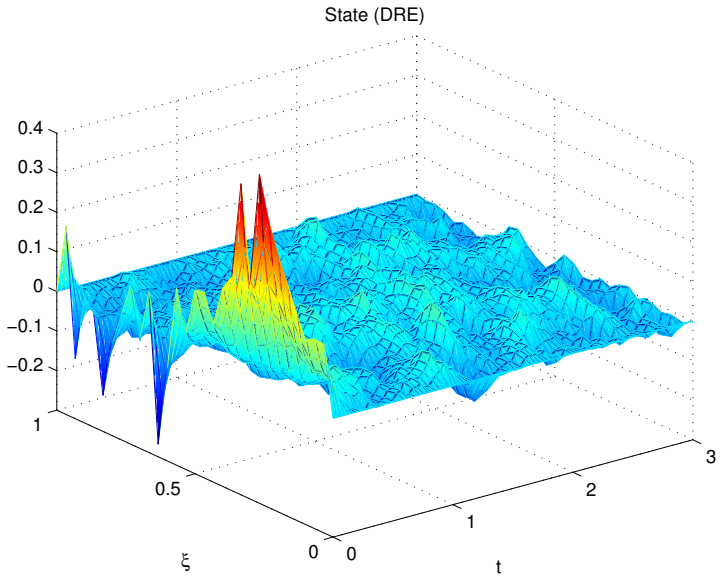


(a)

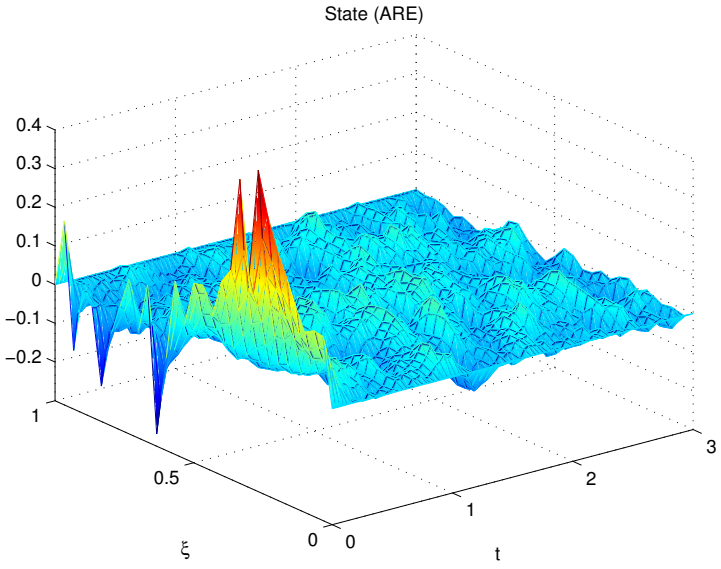


(b)

Figure 6.9: Burgers equation (a) optimal control with noise in the initial condition (DRE) and (b) (ARE) for initial mesh.

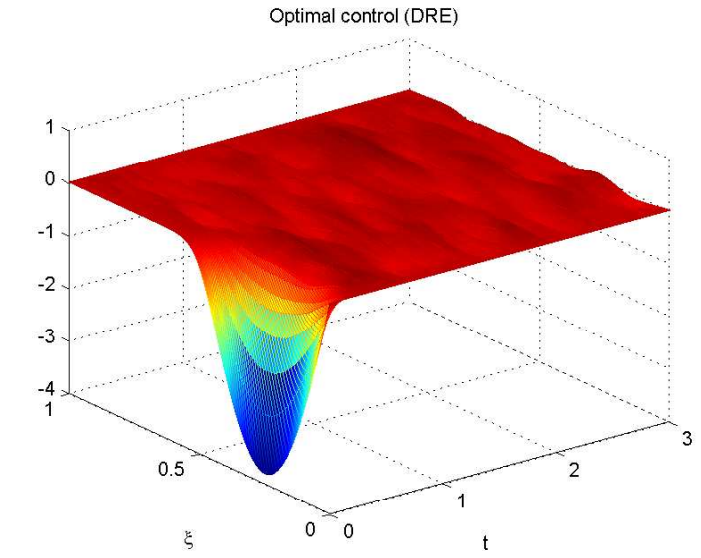


(a)

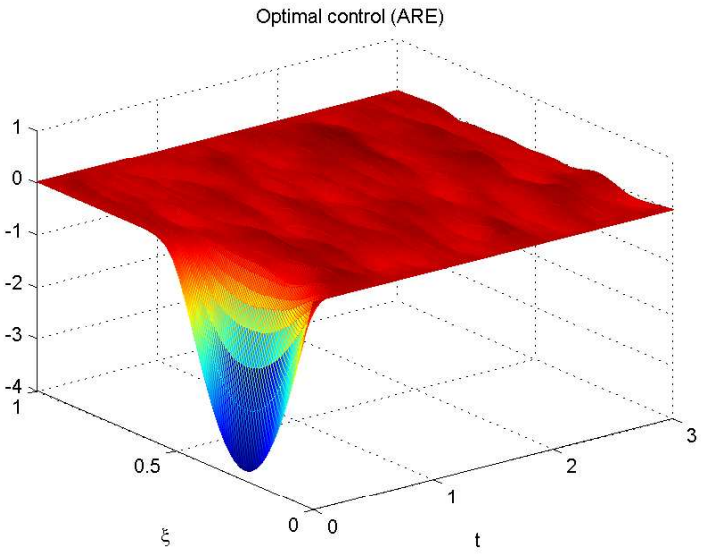


(b)

Figure 6.10: Burgers equation (a) state with noise in the initial condition (DRE) and (b) (ARE) for initial mesh.

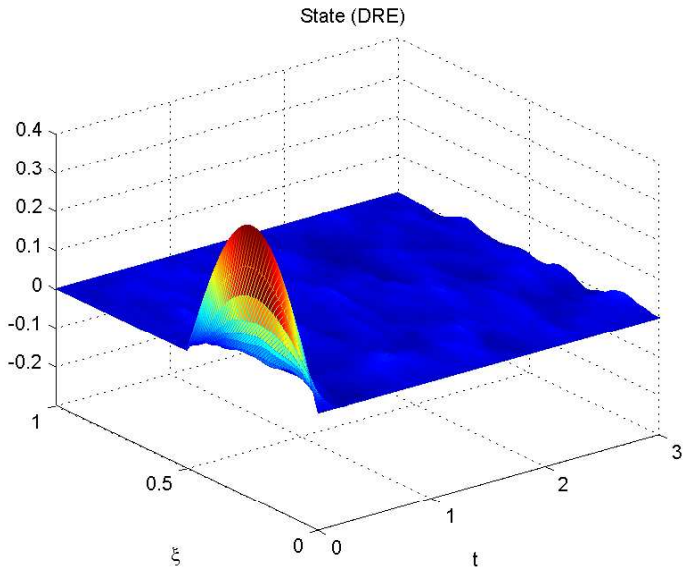


(a)

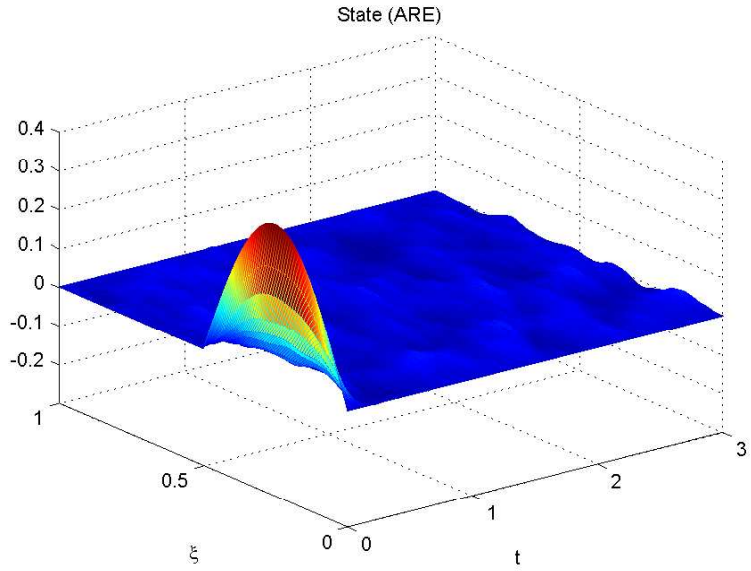


(b)

Figure 6.11: Burgers equation (a) optimal control (DRE) and (b) (ARE) for refined mesh.

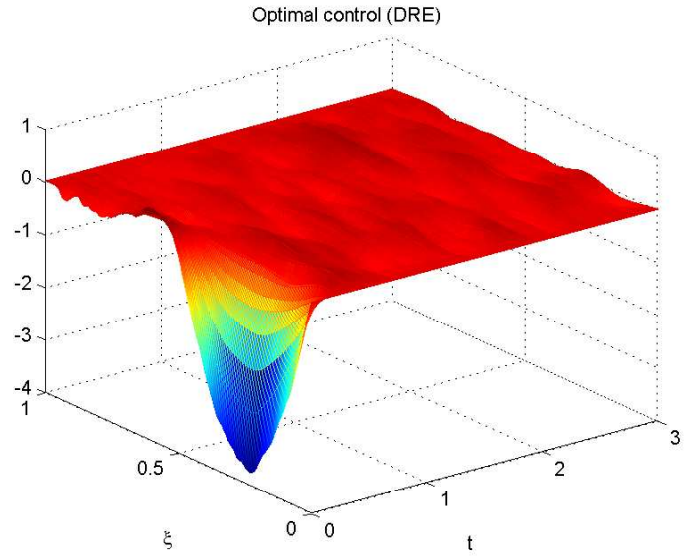


(a)

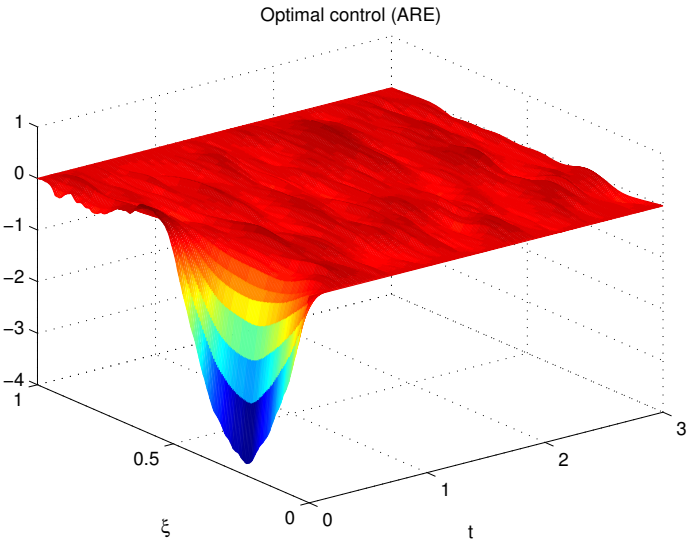


(b)

Figure 6.12: Burgers equation (a) state (DRE) and (b) (ARE) for refined mesh.

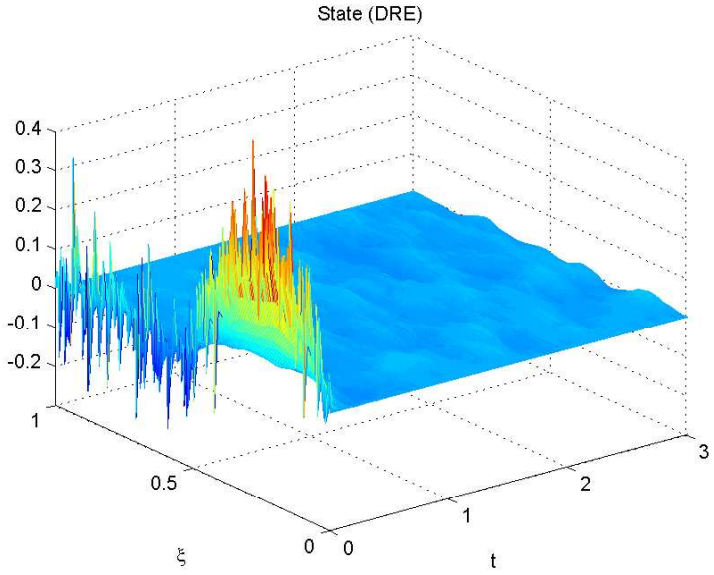


(a)

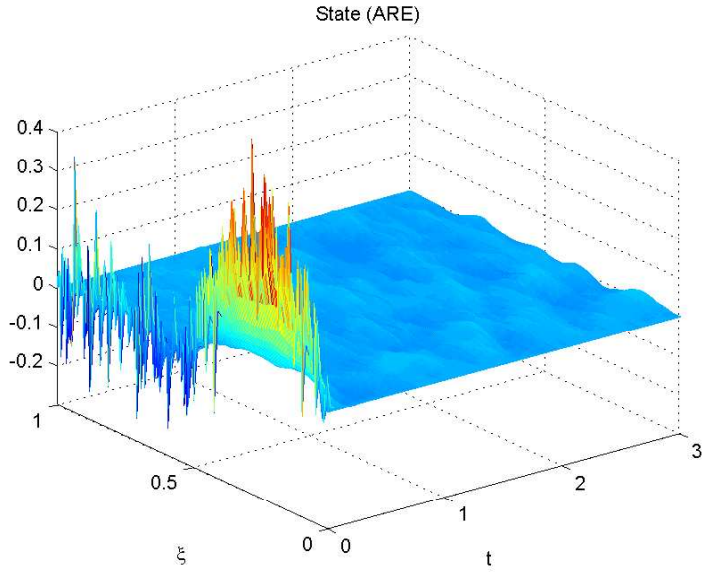


(b)

Figure 6.13: Burgers equation (a) optimal control with noise in the initial condition (DRE) and (b) (ARE) for refined mesh.



(a)



(b)

Figure 6.14: Burgers equation (a) state with noise in the initial condition (DRE) and (b) (ARE) for refined mesh.

CHAPTER

SEVEN

Conclusions and outlook

7.1 Conclusions

The numerical solution of differential Riccati equations (DREs) arising in optimal control problems for parabolic partial differential equations has been the main topic of this thesis. As we have seen the linear-quadratic optimal control problems for partial differential equations on a finite-time horizon immediately leads to the problem of solving large-scale DREs resulting from the semi-discretization using spatial finite element Galerkin scheme. In order to give us an approximation framework for the computation of the infinite-dimensional Riccati equations, in Chapter 3 we have shown the convergence of the finite-dimensional Riccati operators (i.e. the operators related to a matrix DRE) to the infinite-dimensional ones for the autonomous and the non-autonomous case, i.e., the case in which the system is modeled by partial differential equations with time-invariant coefficients and time-varying ones. We also have shown that our result could be extended to other approximation schemes, e.g., spectral methods.

In Chapter 4, we have reviewed the existing methods to solve DREs and investigate whether they are suitable for large-scale problems. We focused on the matrix versions of standard stiff ODE methods. First, in Section 4.2 we concentrated on the BDF methods which are the most popular linear multistep methods for stiff problems. Solving the DRE using BDF methods requires the solution of one ARE in every step. The Newton-ADI iteration is an efficient numerical method for this task. It includes the solution of a Lyapunov equation by a low rank version of the alternating direction implicit (ADI) algorithm in each iteration step. We proposed an efficient implementation for the BDF methods which exploits the given structure of the coefficients matrices. The crucial question of suitable stepsize and order selection strategies is also addressed in terms of the low rank factors of the solution.

Implicit Runge-Kutta methods, or collocation methods, offer an alternative to the BDF methods for stiff problems. Among the implicit Runge-Kutta type

methods which give satisfactory results for stiff problems, e.g. Radau methods, or Gauss and Lobatto methods which extend midpoint and trapezoid rules, the linearly implicit methods (better known as Rosenbrock methods) are the easiest to implement. In fact, as for the BDF methods solving the DRE using midpoint or trapezoid rules requires the solution of an ARE in every step, however there are some technical difficulties which increase the computational cost of solving the ARE by Newton's method in every step, like for instance writing the constant term as a low rank factor product. Instead, an s stage Rosenbrock methods requires only the solution of one Lyapunov equation per stage in every step. Moreover, they posses excellent stability properties (as they can be made A-stable and L-stable). Therefore, we focus on the Rosenbrock methods in Section 4.3. For the case in which the coefficient matrices of the Lyapunov equation are dense, the Bartels-Stewart method can be applied for solving the equations. If the coefficient matrices of the DRE have a certain structure (e.g. sparse, symmetric or low rank, as is the case for DREs arising in optimal control problems which we are interested to solve), the solution of the resulting Lyapunov equation with the Bartels-Stewart method is not feasible. Instead, a low rank version of the ADI algorithm can be applied. We show that it is possible to efficiently implement Rosenbrock methods for large-scale DREs based on this approach.

Due to the fact that, the convergence of the ADI algorithm strongly depends on the set of shift parameters chosen, a new method for determining sets of shift parameters for the ADI algorithm is proposed in Section 4.4. We reviewed existing methods for determining sets of ADI parameters and based on this review we suggest a new procedure which combines the best features of two of those. For the real case, the parameters computed by the new method are optimal and in general their performance is quite satisfactory as one can see in the numerical examples. The computational cost depends only on an Arnoldi process for the matrix involved and on the computation of elliptic integrals. Since the latter is a quadratically converging scalar iteration, the Arnoldi process is the dominant computation here, which makes this method suitable for the large-scale systems arising from finite element discretizations of PDEs. The main advantages of the new method are, that it is cheaper to compute than the existing ones and that it avoids complex computations in the ADI iteration for many cases where the others would result in complex iterations. The efficiency of our method have been shown in Section 4.4.4.

The utility of the Rosenbrock as well as the BDF methods has been demonstrated by numerical experiments in Chapter 5. We want to apply our method to large-scale problems where higher order methods are not feasible to apply due to the computational cost and memory requirements. Furthermore, in large applications fixed step size solvers seem to be more practical, and cheaper to compute, than variable step size ones. This relies on the fact that variable step size solvers are quite sensitive to initial transients and therefore can require rather small step sizes to start up the integrator. Therefore, even though we have controlled the step size directly for the low rank factors for BDF and Rosenbrock methods the computational cost for large-scale problems is still high. If a

variable step size solver has to be applied, then the Rosenbrock method of order two is a reasonable option for the autonomous case. Note that for the non-autonomous case, the computational cost of the Rosenbrock method increases considerably due to the approximation of the derivative involved, here the BDF methods are the better option.

The computational cost and memory requirements for solving the optimal control problems considered in this thesis are high, particularly for nonlinear problems in which several DREs have to be solved. Therefore, the solution of the DREs by higher order methods, or by a variable step size method is still not suitable. For the autonomous case the linearly implicit Euler method (Rosenbrock method of order one) currently appears to be the best option. However, the derivative involved for the non-autonomous case makes the method computationally more expensive. Thus, the implicit Euler method is the better option here.

7.2 Opportunities for future research

Regarding the numerical solution of DREs arising in optimal control problems for parabolic PDEs there remain a number of open questions. The high computational cost of solving control problems suggest to use the resources of processors to deal with large-scale applications, hence parallelization of the methods proposed here is the next step in our research. A parallel solution of large-scale generalized AREs based on the Newton-ADI iteration has been proposed recently, [9]. On the other hand, the memory requirement can be drastically reduce storing the data just in selected points. The selection procedure may be performed applying checkpoint techniques. An memory efficient numerical solution of the control problems we have considered should apply this reduction of storage technique. The method has already proved to be effective for ODE constrained optimal control problems, [109].

In the context of numerical methods to solve DREs, the application of the linearization method to solve DREs has to be investigated further. As we reviewed in Section 4.1, it requires the computation of e^H, where H is the Hamiltonian matrix associated to the DRE. If we approximate e^H by $Ve_k^HV^T$, where $\text{range}(V) = \text{span}\{x, Hx, \ldots, H^{n-1}x\}$, $k \ll n$, then the method could be applied to large-scale DREs.

The solution of the DREs by the BDF methods requires the solution of one ARE in every step. In case the matrix A is stable the initial stabilizing point for solving the first ARE by Newton-ADI iteration can be chose equal zero. If not, choosing the initial stabilizing point for solving the first ARE by Newton-ADI iteration can be computed following [60, 100].

We study here an L-stable second order Rosenbrock method which gives satisfactory results. Higher order Rosenbrock methods for solving DREs have to be investigated further.

Throughout this thesis we have worked in real arithmetics. That is why we skip a comparison of the shift parameters for the ADI iteration in case

they are complex. Particularly, it will be interesting to analyze the behavior of generalized Leja points (which are asymptotically optimal) for the case in which the Wachspress approach is no longer applicable or a-priori information on the spectrum is known.

Finally, we point out that an error estimator from the finite element discretization which controls the whole approach, has to be investigated. That will provide a complete mathematical framework to solve the linear problems. Besides this error estimator, for nonlinear problems a criterion to chose the size of the time frames have to be found.

APPENDIX

A

Stochastic processes

Basic concepts

Let us consider a random variable $J(t)$ depending on the parameter t, then $J(t)$ is called a stochastic process.
The autocovariance for this process is given by

$$\Phi_{JJ}(t_1, t_2) = \mathbb{E}[(J(t_1) - \mathbb{E}[J(t_1)])(J(t_2) - \mathbb{E}[J(t_2)])].$$

If the stochastic properties are invariant with respect to time shifts, that is $J(t) = J(t + c)$ for all t and the expected value $\mathbb{E}[J(t)] = \eta_J$ is constant. Then we have,

$$\Phi_{JJ}(\tau) = \mathbb{E}[(J(t) - \eta_J)(J(t + \tau) - \eta_J)].$$

Definition A.0.1 *A stochastic process is called white noise if $J(t_1)$ and $J(t_2)$ are stochastic independent for all $t_1 \neq t_2$ and the expected value is 0, i.e. $\mathbb{E}[J(t)] = 0$.*

Then, for a white noise, we have

$$\Phi_{JJ}(\tau) = \Phi_0 \delta(\tau),$$

where

$$\Phi_0 > 0, \quad \text{and} \quad \delta(\tau) = \begin{cases} 1 & \text{for } \tau = 0, \\ 0 & \text{otherwise.} \end{cases}$$

In case of a vectorial stochastic process $J(t) = [\, J_1(t), \ldots, J_m(t) \,]$ we obtain a time-dependent covariance matrix

$$\Phi_0(t) = \operatorname{cov}(J(t)) = \mathbb{E}[J(t)J(t)^T] = \begin{bmatrix} \mathbb{E}[J_1(t)J_1(t)] & \ldots & \mathbb{E}[J_1(t)J_m(t)] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[J_m(t)J_1(t)] & \ldots & \mathbb{E}[J_m(t)J_m(t)] \end{bmatrix}$$

If we assume that $J_i(t)$ and $J_j(t)$ are uncorrelated, then all non-diagonal elements are zero. After time discretization we obtain a diagonal covariance matrix for every t_i. Using the same model and measurement-tool over the time horizon, we can assume that the covariance matrices are time-independent.
In Section 6.3, we denote by $V$and W the covariance matrices for the noise processes $v(t)$ and $w(t)$.

BIBLIOGRAPHY

[1] J. Abels and P. Benner. CAREX - a collection of benchmark examples for continuous-time algebraic Riccati equations. Technical report, 1999. SLICOT working note 1999-14. Available from `http://www.slicot.org`.

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

[3] M. Abramovitz and I.A. Stegun, editors. *Pocketbook of mathematical functions*. Verlag Harry Deutsch, 1984. Abridged edition of "Handbook of mathematical functions" (1964).

[4] F. Allgöwer, T. Badgwell, J. Qin, J. Rawlings, and S. Wright. Nonlinear predictive control and moving horizon estimation-An introductory overview. In ed. P. Frank, editor, *in Adavances in Control*, pages 391–449, London, 1999. Springer.

[5] B.D.O. Anderson and J.B. Moore. *Linear Optimal Control*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[6] 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.

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

[8] M. Athans and P.L. Falb. *Optimal Control*. McGraw-Hill, New York, 1966.

[9] 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.

[10] T. Bagby. On interpolation by rational functions. *Duke Math. J.*, 36:95–104, 1969.

[11] H.T. Banks and K. Ito. A numerical algorithm for optimal feedback gains in high dimensional linear quadratic regulator problems. *SIAM J. Cont. Optim.*, 29(3):499–515, 1991.

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

[13] R.H. Bartels and G.W. Stewart. Solution of the matrix equation AX+XB=C: Algorithm 432. *Comm. ACM*, 15:820–826, 1972.

[14] P. Benner. Computational methods for linear-quadratic optimization. *Supplemento ai Rendiconti del Circolo Matematico di Palermo, Serie II*, No. 58:21–56, 1999.

[15] P. Benner. Efficient algorithms for large-scale quadratic matrix equations. *Proc. Appl. Math. Mech.*, 1(1):492–495, 2002.

[16] P. Benner. Solving large-scale control problems. *IEEE Control Systems Magazine*, 14(1):44–59, 2004.

[17] P. Benner and S. Görner. MPC for the Burgers equation based on an LGQ design. *Proc. Appl. Math. Mech.*, 6(1):781–782, 2006.

[18] P. Benner, S. Görner, and J. Saak. Numerical solution of optimal control problems for parabolic systems. In K.H. Hoffmann and A. Meyer, editors, *Parallel Algorithms and Cluster Computing. Implementations, Agorithms, and Applications*, Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin/Heidelberg, Germany, 2006.

[19] P. Benner, J.R. Li, and T. Penzl. Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems. *unpublished manuscript*, 1999.

[20] 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.

[21] 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.

[22] P. Benner, H. Mena, and J. Saak. On the parameter selection problem in the Newton-Adi iteration for large-scale Riccati equations. *to appear in ETNA*.

[23] P. Benner and J. Saak. Efficient numerical solution of the LQR-problem for the heat equation. *Proc. Appl. Math. Mech.*, 4(1):648–649, 2004.

[24] P. Benner and J. Saak. Linear-quadratic regulator design for optimal cooling of steel profiles. Technical Report SFB393/05-05, Sonderforschungsbereich 393 *Parallele Numerische Simulation für Physik und Kontinuumsmechanik*, TU Chemnitz, D-09107 Chemnitz (Germany), 2005. Available from `http://www.tu-chemnitz.de/sfb393/sfb05pr.html`.

[25] P. Benner and J. Saak. A semi-discretized heat transfer model for optimal cooling of steel profiles. In P. Benner, V. Mehrmann, and D. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, Lecture Notes in Computational Science and Engineering, pages 353–356. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

[26] 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.

[27] 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.

[28] N.P. Bhatia and G.P. Szegö. *Stability Theory of Dynamical Systems.* Classics in Mathematics. Springer-Verlag, Berlin Heidelberg, 2002.

[29] 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.

[30] 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.

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

[32] P.N. Brown, G.D. Byrne, and A.C. Hindmarsh. VODE: a variable coefficient ode solver. *SIAM J. Sci. Stat. Comput*, 10:1039–1051, 1989.

[33] J. Bruder. Numerical results for a parallel linearly-implicit runge-kutta method. *Computing*, 59(2):139–151, 1997.

[34] R.S. Bucy and R.E. Kalman. New results in linear filtering and prediction theory. *Trans. ASME, Series D*, 83:95–108, 1961.

[35] J.L. Casti. *Linear Dynamical Systems.* Academic Press, New York, 1987.

[36] Y. Chahlaoui and P. Van Dooren. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. SLICOT Working Note 2002–2, February 2002. Available from `http://www.slicot.org`.

[37] C. Choi and A.J. Laub. Constructing Riccati differential equations with known analytic solutions for numerical experiments. *IEEE Trans. Automat. Control*, 35:437–439, 1990.

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

[39] 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.

[40] 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.

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

[42] R. Datko. A linear control problem in abstract Hilbert space. *J. Diff. Eqns.*, 9:346–359, 1971.

[43] R. Datko. Unconstrained control problem with quadratic cost. *SIAM J. Control*, 11:32–52, 1973.

[44] E.D. Davies. *One parameter semigroups*. Academic Press, 1980.

[45] E.J. Davison and M.C. Maki. The numerical solution of the matrix Riccati differential equation. *IEEE Trans. Automat. Control*, 18:71–73, 1973.

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

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

[48] K-L. Engel and R. Nagel. *One-Parameter Semigroups for Linear Evolution Equations*. Springer-Verlag, New York, 2000.

[49] K. Eppler and F. Tröltzsch. Discrete and continuous optimal control strategies in the selective cooling of steel profiles. *Z. Angew. Math. Mech.*, 81:247–248, 2001.

[50] A.D. Freed and I.s. Iskovitz. Development and applications of a Rosenbrock integrator. Technical Report NASA Tech. Memorandum 4709, Lewis Research Center, Cleveland, Ohio, 1996.

[51] C.E. Garcia, D.M. Prett, and M.Morari. Model predictive control: Theroy and practice–a survey. *Automatica*, 25:335–348, 1989.

[52] J.S. Gibson. The Riccati integral equation for optimal control problems in Hilbert spaces. *SIAM J. Cont. Optim.*, 17(4):537–565, 1979.

[53] J.A. Goldstein. *Semigroups of Operators and Applications*. Oxdorf Universty Press, 1985.

[54] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[55] A. A. Gonchar. Zolotarev problems connected with rational functions. *Math USSR Sbornik*, 7:623–635, 1969.

[56] 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.

[57] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I-Nonstiff Problems*. Springer Series in Computational Mathematics. Springer-Verlag, New York, 2000.

[58] 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.

[59] J. Harnard, P. Winternitz, and R. L. Anderson. Superposition principles for matrix Riccati equations. *J. Math. Phys.*, 24:1062–1072, 1983.

[60] C. He and V. Mehrmann. Stabilization of large linear systems. In L. Kulhav'a, M. K'arn'y, and K. Warwick, editors, *Preprints of the European IEEE Workshop CMP'94*, pages 91–100, 1994.

[61] E. Hille and R.S. Phillips. *Functional Analysis and Semigroups*, volume vol. 31. American Mathematical Society, 1957.

[62] A.C. Hindmarsh. LSODE and LSODI: two new initial value ordinary differential equation solvers. *ACM-SIGNUM Newsletter*, 15:10–11, 1980.

[63] A. Ichikawa and H. Katayama. Remarks on the time-varying H_∞ Riccati equations. *Sys. Cont. Lett.*, 37(5):335–345, 1999.

[64] M. P. Istace and J. P. Thiran. On the third and fourth Zoltarev problems in complex plane. *SIAM J. Numer. Anal.*, 32(1):249–259, 1995.

[65] K. Ito. Finite-dimensional compensators for infinite-dimensional systems via Galerkin-type approximation. *SIAM J. Cont. Optim.*, 28:1251–1269, 1990.

[66] K. Ito and K. Kunisch. Asymptotic properties of receding horizon optimal control problems. *SIAM J. Cont. Optim.*, 40(5):1585–1610, 2002.

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

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

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

[70] C. Kenney and R.B. Leipnik. Numerical integration of the differential matrix Riccati equation. *IEEE Trans. Automat. Control*, AC-30:962–970, 1985.

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

[72] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations Analysis and Numerical Solution*. EMS Publishing House, Zürich, Switzerland, 2006.

[73] D.G. Lainiotis. Generalized Chandrasekhar algorithms: Time-varying models. *IEEE Trans. Automat. Control*, AC-21:728–732, 1976.

[74] P. Lancaster and L. Rodman. *The Algebraic Riccati Equation*. Oxford University Press, Oxford, 1995.

[75] 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.

[76] 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.

[77] 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.

[78] A.J. Laub. Schur techniques for Riccati differential equations. In D. Hinrichsen and A. Isidori, editors, *Feedback Control of Linear and Nonlinear Systems*, pages 165–174, New York, 1982. Sringer-Verlag.

[79] V. I. Lebedev. On a Zolotarev problem in the method of alternating directions. *USSR Comput. Math. and Math. Phys.*, 17:58–76, 1977.

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

[81] X. Li and J. Yong. *Optimal Control Theory for Infinite Dimensional Systems*. Birkhäuser, Boston, 1995.

[82] J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971.

[83] J.L. Lions and E. Magenes. *Non-Homogeneous Boundary Value Problem, I, II, III*. Springer-Verlag, New York, 1972.

[84] A. Locatelli. *Optimal Control*. Birkhäuser, Basel, Boston, Berlin, 2001.

[85] A. Lu and E.L. Wachspress. Solution of Lyapunov equations by alternating direction implicit iteration. *Comput. Math. Appl.*, 21(9):43–58, 1991.

[86] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*. Number 163 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Heidelberg, July 1991.

[87] K.A. Morris. Convergence of controllers designed using state-space methods. *IEEE Trans. Automat. Control*, 39:2100–2104, 1994.

[88] K.A. Morris. Design of finite-dimensional controllers for infinite-dimensional systems by approximation. *J. Math. Systems, Estim. and Control*, 4:1–30, 1994.

[89] A. Pazy. *Semigroups of Linear Operators and Applications to Partial Differential Equations*, volume 44 of *Appl. Math. Sci.* Springer-Verlag, 1983.

[90] D.W. Peaceman and H.H. Rachford Jr. The numerical solution of parabolic and elliptic differential equations. *J. SIAM*, 3:28–41, 1955.

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

[92] T. Penzl. A cyclic low rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 2000.

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

[94] T. Penzl. LYAPACK Users Guide. Technical Report SFB393/00-33, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, 09107 Chemnitz, Germany, 2000. Available from `http://www.tu-chemnitz.de/sfb393/sfb00pr.html`.

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

[96] P.H. Petkov, N.D. Christov, and M.M. Konstantinov. *Computational Methods for Linear Control Systems*. Prentice Hall, Hertfordshire, UK, 1991.

[97] 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.

[98] E. R. Pinch. *Optimal Control and the Calculus of Variations*. Oxford University Press, Oxford, UK, 1993.

[99] 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.

[100] X. Rao. Large scale stabilization with linear feedback, Master Thesis, Florida State University, Deparment of Computer Science, Fall Semester 1999.

[101] W.T. Reid. *Riccati Differential Equations*. Academic Press, New York, 1972.

[102] D.L. Russell. *Mathematics of Finite-Dimensional Control Systems*, volume 43 of *Lecture Notes in Pure and Applied Mathematics*. Marcel Dekker Inc., New York, 1979.

[103] 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.

[104] L. F. Shampine and M.K. Gordon. *Computer Solution of Ordinary Differential Equations*. Freeman, San Francisco, 1975.

[105] V. Sima. *Algorithms for Linear-Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics*. Marcel Dekker Inc., New York, 1996.

[106] E. D. Sontag. *Mathematical Control Theory. Deterministic Finite Fimensional Systems*. in Texts in Applied Mathematics. Springer-Verlag, New York, NY, 2nd. edition, 1998.

[107] G. Starke. *Rationale Minimierungsprobleme in der komplexen Ebene im Zusammenhang mit der Bestimmung optimaler ADI-Parameter*. PhD thesis, Fakultät für Mathematik, Universität Karlsruhe, December 1989.

[108] G. Starke. Optimal alternating directions implicit parameters for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.*, 28(5):1431–1445, 1991.

[109] J. Sternberg. *Reduction of storage requirement by checkpointing for time-dependent optimal control problems*. PhD thesis, Technische Universität Dresden, December 2005.

[110] U. Storch and H. Wiebe. *Textbook of mathematics. Vol. 1: Analysis of one variable. (Lehrbuch der Mathematik. Band 1: Analysis einer Veränderlichen.)*. Spektrum Akademischer Verlag, Heidelberg, 3 edition, 2003. (German).

[111] J. Todd. Applications of transformation theory: A legacy from Zolotarev (1847-1878). In S. P. Singh et al., editor, *Approximation Theory and Spline Functions*, number C 136 in NATO ASI Ser., pages 207–245, Dordrecht-Boston-Lancaster, 1984. D. Reidel Publishing Co.

[112] 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.

[113] D. A. Voss and A. Q. M. Khaliq. Parallel Rosenbrock methods for chemical systems. *Computers and Chemestry*, 25:101–107, 2001.

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

[115] E.L. Wachspress. The ADI model problem, 1995. Available from the author.

[116] D. R. Willé. New stepsize estimators for linear multistep methods. Technical Report Numerical Analysis No. 247, University of Manchester/UMIST, Manchester M13 9PL, 1994.

[117] W. M. Wonham. *Linear multivariable control*. Lecture Notes in Economics and Mathematical Systems 101. Spektrum Akademischer Verlag, New York, 1974.

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

INDEX