Extension of inexact Kleinman-Newton methods to a general monotonicity preserving convergence theory

Dissertation

zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)

Dem Fachbereich IV der Universität Trier
vorgelegt von

Timo Hylla

Trier, 2010

Meiner lieben Frau Catherine gewidmet.

Dissertationsort: Trier

Danksagung

Ich möchte mich ganz herzlich bei all jenen Personen bedanken, die zum Erstehen dieser Dissertation beigetragen haben.

Mein tiefster Dank richtet sich an Herrn Prof. Dr. Sachs, der es mir ermöglichte und mich dazu ermutigte das Projekt "Promotion" in Angriff zu nehmen. Dank seiner Person wurde die Arbeit an der Universität niemals als solche wahrgenommen. Fachlich war er mir eine große Hilfe und er gab mir die Gelegenheit, an zahlreichen Tagungen und Fortbildungen teilzunehmen. Auch über die Arbeit hinaus war und ist er mir immer ein wichtiger Ansprechpartner.

Herzlich bedanken möchte ich mich auch bei meinen lieben Kolleginnen und Kollegen, die mittlerweise zu guten Freunden geworden sind. Mit vielen erholsamen Kaffeepausen und fachlichen Gesprächen wurde ein wichtiger Beitrag zum Fertigstellen der Arbeit geleistet.

Schließlich gebührt mein Dank meiner Familie. Meine Eltern haben mir das Studium erst ermöglicht und standen mir immer mit Rat und Tat zur Seite. Gemeinsam mit meinem Bruder und seiner Familie boten Sie mir einen Rückzugsort und Ablenkung, wenn mir die Arbeit über den Kopf zu wachsen schien.

Zum Schluss, mir jedoch am wichtigsten, möchte ich mich bei meiner Frau bedanken. Ohne deren Liebe und Beistand wäre diese Arbeit niemals fertig gestellt worden. Ihr kann ich nur das größte Kompliment machen: "Ich würde niemals mein Leben mit jemand anderem tauschen wollen".

Contents

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit inexakten Newton Verfahren, die auf algebraische Riccati Gleichungen angewendet werden. In diesem speziellen Fall liefert das Newton Verfahren monotone Iterierte und konvergiert in einer globaleren Weise als erwartet. Ein zentrales Ziel der Arbeit besteht darin, Voraussetzungen an die inexakte Methode zu entwickeln, um ein äquivalentes Konvergenzverhalten zu gewährleisten. Zudem werden inexakte Verfahren mit einer linearen, superlinearen und quadratischen Rate der lokalen Konvergenz präsentiert. Dadurch, dass man bei dem inexakten Newton Verfahren die einzelnen Newton Schritte häufig früher abbrechen kann, gewinnt man einen großen Vorteil in der benötigten Rechenzeit. Dies wird an verschiedenen Beispielen verdeutlicht.
Ein weiterer Schwerpunkt der Arbeit liegt in der Untersuchung einer alternativen Implementierung der Newton Methode für algebraische Riccati Gleichungen. Diese weist in der Praxis häufig Instabiliäten auf, die mit Hilfe von inexakten Newton Verfahren erklärt werden können.
Die Erkenntnisse, welche aus der Arbeit mit Riccati Gleichungen gewonnen werden, führen zu einer Erweiterung der allgemeinen Konvergenztheorie für das inexakte Newton Verfahren. Unter bestimmten Bedingungen an die betrachtete Funktion und den zugrunde liegenden Raum, sichert diese Erweiterung die monotone Konvergenz der Iterierten und als Konsequenz den größeren Konvergenzradius. Zahlreiche Beispiele erfüllen die Anforderungen der Theorie und werden als Anwendungen aufgeführt. Darunter fallen unter anderem nicht symmetische Riccati Gleichungen, Riccati Gleichungen aus der stochastischen Optimierung und Anwendungen aus der Quasilinearisierungstechnik.

Chapter 1

Introduction

1.1 Motivation and review of the literature

The solution of algebraic Riccati equations is still an ambitious task, especially for equations arising in large scale control systems. These equations are often solved within the framework of so-called Kleinman-Newton methods [45]. In order to reduce computing time in this context, the implementation of iterative solvers for the solution of the linear systems occurring at each Newton step is unavoidable. In such a numerical approach, it is important to control the accuracy of the solution of the linear systems at each Newton iteration in order to gain efficiency, but not to lose the fast convergence properties of Newton's method.
Here inexact Newton's methods would provide a stringent guideline for the termination of the inner iteration, resulting in different rates of local convergence. In addition, the early termination of the iterations bears a striking effect in saving computing time as long as the iterates are still far away from the solution. An interesting discussion on inexact Newton's method can be found in a book from Kelley [44].
In his classical paper, Kleinman [45] applied Newton's method to the algebraic Riccati equation (ARE), a quadratic equation for matrices of the type

$$A^T X + XA - XBB^T X + C^T C = 0 \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{l\times n}$. At each Newton step, a Lyapunov equation

$$X_{k+1}(A - BB^T X_k) + (A - BB^T X_k)^T X_{k+1} = -X_k BB^T X_k - C^T C \tag{1.2}$$

needs to be solved to obtain the next iterate X_{k+1}.
Algebraic Riccati equations play an important role in the solution of time-

invariant linear quadratic regulator (LQR) problems over an infinite time horizon

$$\begin{aligned}\min_{u\in L^2_m(0,\infty)} J(u,x_0) &= \frac{1}{2}\int_0^\infty \left(y(t)^T Q y(t) + u(t)^T R u(t)\right) dt \\ s.t. \quad \dot{x}(t) &= Ax(t) + Bu(t), \qquad t>0, \qquad x(0)=x_0 \\ y(t) &= Cx(t), \qquad\qquad\quad t>0,\end{aligned}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$, $Q \in \mathbb{R}^{l\times l}$ and $R \in \mathbb{R}^{m\times m}$. The solution of LQR problems is a fundamental field in control theory, see e.g. [46, 2, 54, 48]. Under suitable conditions on the system matrices, the optimal control $u_*(t)$ is given by a feedback law, namely

$$u_*(t) = -R^{-1}B^T X_\infty x(t), \qquad t>0, \tag{1.3}$$

where X_∞ is defined as the stabilizing solution of an algebraic Riccati equation

$$A^T X + XA - XBR^{-1}B^T X + C^T Q C = 0.$$

Kleinman [45] introduced his well-known convergence results in 1968, nevertheless the numerical solution of Riccati equation is still a vivid field of research.
Banks and Ito [3] developed a second implementation of the Kleinman-Newton method, where the Newton step is computed by a Lyapunov equation for the increment $X_{k+1} - X_k$ in the following way

$$\begin{aligned}(X_{k+1}-X_k)(A-BB^TX_k) + (A-BB^TX_k)^T(X_{k+1}-X_k) \\ = (X_k - X_{k-1})BB^T(X_k - X_{k-1}).\end{aligned} \tag{1.4}$$

They utilize Chandrasekhar's method for the computation of a stable initial iterate and introduced the first feedback gain algorithm. [55] presents a comparison of both Kleinman-Newton versions. Against one's expectations, the Banks and Ito method does not represent the matrix formulation of a standard Newton step

$$\mathcal{F}'(X_k)(X_{k+1}-X_k) = -\mathcal{F}(X_k).$$

Here the right side $-\mathcal{F}(X_k)$ has been modified due to Taylor expansion and the quadratic nature of the algebraic Riccati equation, see [41] for details. In addition, this paper stated a kind of instability of the modified version, which can be explained and analyzed with help of inexact Newton's method.
Inexact variants of the Kleinman-Newton method have been published in [22]. Benner and Byers [6] followed by Guo and Laub [36] incorporated line searches in a Newton procedure. Both modifications of Newton's method lead to a reduction in the number of inner iterations.
Multilevel techniques for the solution of the Riccati equation have been proposed

by Rosen and Wang [64]. The case of a singular Jacobian at the solution has been analyzed by Guo and Lancaster [35].
Burns, Sachs, and Zietsman [16] give conditions under which the Kleinman-Newton method is mesh independent. Here the number of iterates remains virtually constant when the discretization of the underlying optimal control problem is refined.
Since each step of the Kleinman-Newton method is equivalent to the solution of a corresponding Lyapunov equation, a matrix equation for S of the type

$$AS + SA^T + W = 0,$$

where $A \in \mathbb{C}^{n\times n}$ and $W \in \mathbb{C}^{n\times n}$ are given matrices, all contributions for iterative Lyapunov solvers are also important in case of Kleinman-Newton methods. There is a sizeable amount of literature on how to solve Lyapunov equations with direct solvers and iterative methods. Direct Lyapunov solvers are presented e.g. in Laub [49], Roberts [62] or Grasedyck [29]. Various iterative solvers can be found in [70, 74, 59, 60, 50, 51, 31, 75, 69], where parameter selection procedures play a crucial role, see e.g. [8, 67] and the references therein. [7] provides an interesting discussion on one state-of-the-art Lyapunov solver, namely the low-rank Cholesky factor ADI method. An efficient implementation of this method is provided in the M.E.S.S. (Matrix Equation Sparse Solver) package [9], the successor of the LyaPack Matlab Toolbox [58].
In case of linear quadratic regulator (LQR) problems [46, 2, 54, 48] one is not mainly interested in a solution X_∞ of equation (1.1) only the low-dimensional matrix $B^T X_\infty$ is of practical importance. Feedback gain algorithms, see e.g. [3, 7], take advantage of this fact and improve the performance of the existing algorithms. Until now it has not been considered how those feedback gain algorithms can be applied in an inexact Newton context.
All Newton iterates, defined in (1.2), show a monotone convergence property, i.e. $X_k \geq X_{k+1}$ for $k \geq 1$, which is not common for Newton's method. A more global convergence can be stated due to this monotonicity of the iterates. As a result, the initial iterate does not have to lie in a neighborhood of the solution.
In several other applications Newton's method shows a similar convergence property. Here one could mention nonsymmetric Riccati equation [25], Riccati equation in stochastic control [76] and a general matrix equation introduced by Damm and Hinrichsen [19, 18].
These phenomena have been analyzed for Newton's method e.g. in [56]. In case of inexact Newton's methods, the question of monotonicity has not been considered before.

1.2 Summary of the thesis

In this thesis we discuss inexact Newton methods in several areas of application and place special emphasis on a monotone convergence property. Our work is initiated with an analysis of the applicability of inexact Newton methods in the context of algebraic Riccati equations [41, 22] and is concluded in a general monotonicity preserving convergence theory for inexact Newton methods. Due to the monotonicity of the inexact Newton iterates, we are able to state a more global convergence result.

Kleinman [45] applied Newton's method to the solution of algebraic Riccati equations (AREs), already mentioned in (1.1). The goal is to find a symmetric matrix $X \in \mathbb{R}^{n\times n}$ with $\mathcal{F}(X) = 0$, where the nonlinear map $\mathcal{F} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is defined by

$$\mathcal{F}(X) = A^T X + XA - XBB^T X + C^T C.$$

Kleinman stated a remarkable convergence result in this case. Here the initial iterate X_0 is not required to lie in a neighborhood of the solution, only the stability of $A - BB^T X_0$ is necessary. In addition, a monotone convergence of the Newton iterates, i.e. $X_k \leq X_{k+1}$ for all $k \geq 1$, can be observed and proven. These characteristics are not common for Newton's method and depend on the special structure of the ARE.

For large scale systems, the occurring Newton steps can be solved only with the help of iterative solvers. We introduce inexact Kleinman-Newton methods, where an error of size R_k is allowed in the k-th inexact Newton step

$$\begin{gathered}\mathcal{F}'(X_k)(X_{k+1} - X_k) + \mathcal{F}(X_k) = R_k \\ \Longleftrightarrow \\ X_{k+1}(A - BB^T X_k) + (A - BB^T X_k)^T X_{k+1} = -X_k BB^T X_k - C^T C + R_k.\end{gathered}$$

Our theory provides stringent guidelines for the termination of the inner iteration, resulting in different rates of local convergence. Depending on the stopping criterion, restricting the size of $\|R_k\|$ in dependence on the actual iterate, the inexact versions show a linear, superlinear or even quadratic rate of local convergence.

In Figure 1.1 we present some exemplary numerical results to indicate the benefits of the newly developed inexact Kleinman-Newton methods. Here we compare the number of inner iterations for exact and inexact Kleinman-Newton methods, required for the solution of an ARE arising in optimal control problems. The inexact version, providing a superlinear rate of local convergence, computes the solution with significantly fewer steps within the iterative solver. The possibility to terminate the inner iteration early as long as the iterates are far away from the solution leads to a substantial reduction of the numerical effort. We discuss

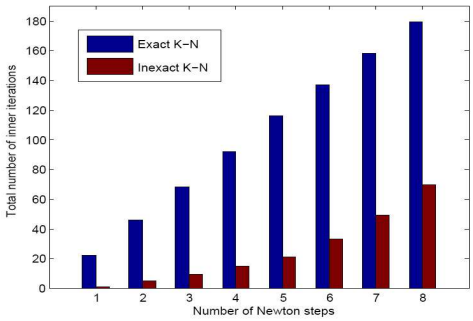

Figure 1.1

the convergence properties of the inexact Kleinman-Newton methods for several examples, taken from linear quadratic regulator (LQR) problems [46].
So-called feedback gain algorithms play an important role in the solution of LQR problems. Those algorithms calculate the iterates $B^T X_k$, without knowledge of the original Newton iterates $X_k,\ k \in \mathbb{N}$. The computation of the Newton iterates can be omitted because the optimal control u_* of a LQR problem, already defined in (1.3), can be calculated as long as the matrix $B^T X_\infty$ is known. Therefore it is possible to work on the low rank iterates $B^T X_k$, which results in a reduction of the numerical effort.
In combination with inexact Newton methods, some difficulties are encountered. In the inexact context, the stopping criteria always require the computation of $\mathcal{F}(X_k)$ and R_k in each Newton step. Both quantities depend on X_k, which is not known in a feedback gain algorithm and can therefore not be evaluated directly. We present alternative representation of $\mathcal{F}(X_k)$ and R_k, which can be computed without knowledge of X_k, only information about $B^T X_k$ is necessary.
An important feature of the Kleinman-Newton method is a global convergence result. The initial iterate X_0 is chosen, such that $A - BB^T X_0$ describes a stable matrix but the closeness of X_0 to the solution is of no importance. In order to retain Kleinman's convergence results, including monotonicity and a global convergence property, also for inexact Kleinman-Newton methods, we have to impose several conditions on the residual R_k of the k-th inexact Newton step. There exists two different types of conditions.
One key assumption is the non-negative definiteness of the residuals $R_k,\ k \in \mathbb{N}$. Since our iterates are computed with an iterative solver, we analyze the most popular iterative Lyapunov solver with respect to their capability to provide non-negative definite residuals.
We stated an important result for the ADI (Alternating Implicit Direction)

method [57, 52], which is also valid for Smith's method [70] and the low-rank Cholesky factor ADI method [7]. The choice of the zero matrix as an initial iterate for these iterative solver is common practice. In addition, the zero matrix leads to a non-negative definite residual and if the initial iterate of the inner iteration provides a non-negative definite residual then all subsequent iterates will also contribute non-negative definite residuals.
The other requirements, e.g. $R_k \leq C^T C$, involve matrix inequalities, which can be tested during the iteration with an additional numerical effort.
As a result, we are able to state a global convergence property under suitable conditions also for the inexact case. This conclusion is important for many Riccati solvers. LyaPack [58] and M.E.S.S. [9] utilize among others some inexact stopping criteria, based an relative changes of the residuals or stagnation techniques. Our theory indicates, that these algorithms calculate the maximal stabilizing solution of the ARE, which is the solution of practical interest.
Furthermore, we analyze an alternative implementation of the Kleinman-Newton method (1.4), introduced by Banks and Ito [3]. Both versions of Newton's method for ARE differ quite substantially, e.g. the right side of (1.4) is independent of C and usually of low numerical rank.
We demonstrate, that inexact Newton methods

$$\begin{aligned}(X_{k+1} - X_k)(A - BB^T X_k) + (A - BB^T X_k)^T (X_{k+1} - X_k) \\ = (X_k - X_{k-1})BB^T (X_k - X_{k-1}) + R_k.\end{aligned}$$

not can be applied successfully in this case. Nevertheless, our considerations explain instabilities, which occured in practice [41]. Newton's method is no longer self-correcting for this modification of the Kleinman-Newton method and this version of an implementation should be handled carefully.
Many equations are closely related to the algebraic Riccati equation (1.1), e.g. nonsymmetric Riccati equation [25], Riccati equation in stochastic control [76] and a general matrix equation introduced by Damm and Hinrichsen [19, 18]. All these examples show similar convergence properties for Newton's method and therefore the application of inexact Newton methods seems very promising. Instead of introducing inexact Newton methods for each single application, we develop a general monotonicity preserving convergence theory, which covers all mentioned examples as special cases.
We establish a sufficient theoretical background to provide conditions on a mapping $\mathcal{F}$ and the residuals $R_k,\ k \in \mathbb{N}$ to secure a monotone convergence of the inexact Newton iterates. As a result we can state a more global convergence property as usual.
In a general framework, an inexact Newton step is defined by

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) = -\mathcal{F}(X_k) + R_k, \tag{1.5}$$

where $\mathcal{F} : D \subset E \to F$, E and F are Banach spaces, D is an open convex subset of E.
All convergence results are based on a specific quality of the Banach space E. It is required, that the monotonicity and boundedness of a sequence $\{x_k\}_{k\in\mathbb{N}} \in E^{\mathbb{N}}$ induce its convergence. For some spaces this is a trivial fact but not for general Banach spaces. Here the concept of a regular proper convex cone [73] secures relation between monotonicity, boundedness and convergence mentioned above.
Of course, the monotonicity of the inexact Newton iterates is also dependent on the structure of the mapping $\mathcal{F}$. On the one hand, we assume $\mathcal{F}$ to be a convex or concave mapping. On the other hand we require a property of its derivative, which can be described within the theory of inverse negativity (positivity).
These requirements are restrictive but many important applications fit to this theory. We show in detail, that nonsymmetric Riccati equation [25] and a general matrix equation, introduced in [19, 18], can be applied in the newly developed monotonicity preserving convergence theory. Since this general matrix equation covers several important equations, e.g. discrete algebraic Riccati equations [48] and Riccati equation occuring in stochastic control [76], the practical benefits of our theory is obvious.
In addition, we analyze the Quasilinearization technique, introduced in [4], with respect to our theory. This method can be interpreted as Newton's method for a nonlinear differential operator equation [63]. We apply this idea to parabolic partial differential equations (PDEs) of the type

$$u_t = u_{xx} + \varphi(u) - f(t,x) \quad \forall (t,x) \in Q_T := (0,T] \times (a,b)$$

with initial and boundary conditions

$$\begin{aligned} u(0,x) &= \tilde{u}(x) && \forall x \in \Omega := (a,b) \\ u(t,x) &= g(t,x) && \forall (t,x) \in \Sigma_T := \{(t,x) |\ t \in (0,T],\ x \in \{a,b\}\}. \end{aligned}$$

Here one defines a function

$$\mathcal{F}(u) := \begin{pmatrix} u_{xx} - u_t + \varphi(u) - f(t,x) \ \ \forall (t,x) \in Q_T \\ \tilde{u}(x) - u(0,x) \quad \forall x \in \Omega \\ g(t,x) - u(t,x) \ \ \forall (t,x) \in \Sigma_T \end{pmatrix} \tag{1.6}$$

and calculates a solution of $\mathcal{F}(u) = 0$ with Newton's method, which is also a solution of the corresponding PDE.
Utilizing the maximum principle and some restrictions on the occuring mappings $\varphi,\ f,\ \tilde{u}$ and g, we show that the function $\mathcal{F}$ satisfies all requirements of the new convergence theory for inexact Newton methods. Since Newton's method can be rarely realized in infinite dimensional function spaces, see e.g. [21], we restrict ourselves to the discretized version of the PDE. Nevertheless our considerations of the infinite dimensional function space indicate the applicability of our theory and therefore we expect the discretized version to behave likewise.

1.3 Outline

The contents of the thesis are organized as follows. Chapter 2 analyzes the applicability of inexact Newton's methods in the context of algebraic Riccati equations. Depending on the stopping criteria, we introduce inexact Kleinman-Newton methods, which provide different rates of local convergence. Kleinman's [45] well-known convergence results, including monotonicity and global convergence property, are extended to inexact Newton's methods. We also study a second implementation of Newton's method for algebraic Riccati equations [3] and explain occurring instabilities.
In chapter 3 we present a review of the most common iterative Lyapunov solvers. We consider whether those iterative solvers fit the monotonicity preserving convergence theory.
Chapter 4 indicates the benefits of the inexact Kleinman-Newton methods. We utilize several numerical examples to compare the convergence properties of the exact Kleinman-Newton method and different inexact versions, with a linear, superlinear or quadratic rate of local convergence. All test examples arise in the context of optimal control problems, especially in linear quadratic regulator problems.
In chapter 5 we briefly discuss feedback gain algorithms. In order to introduce inexact Newton's methods in this context several difficulties are encountered. We outline these problems and initiate some ideas to circumvent them.
Chapter 6 introduces a general convergence theory for inexact Newton's methods. We outline conditions on the mapping and the residuals of an inexact Newton's methods, which secure a monotone and hence more global convergence property. In addition, in chapter 7 we present several important areas of application of the new developed theory.
Finally, chapter 8 summarizes the statements of the thesis and gives some closing remarks.

Chapter 2

Numerical solution of Riccati equations

A major part of this thesis analyzes the applicability of inexact Newton's methods in the context of Riccati equations. In his classical paper, Kleinman [45] applied Newton's method to the algebraic Riccati equation

$$A^T X + XA - XBB^T X + C^T C = 0.$$

Chapter 2.1 reviews his well known convergence results, including monotonicity and global convergence.
We introduce inexact Kleinman-Newton methods in section 2.2. Utilizing standard theorems about inexact Newton's methods, we are able to state local convergence results. Our theory provides stopping criteria resulting in inexact versions with a linear, superlinear or even quadratic rate of convergence.
Under suitable conditions on the initial iterate Newton's method shows a monotone convergence property, which secures a more global convergence as usual. In the inexact case we have to impose several restrictions on the residuals to restore the monotonicity of the iterates and therefore the more global convergence. These convergence results are summarized in section 2.3.
A second formulation of Newton's method for the solution of the ARE has been introduced in the literature, e.g. [3]. In practice, this implementation shows some unexplainable instabilities [41]. Interpreting this version as an inexact Newton's method enables us to understand and analyze the instability. A detailed discussion on this topic is presented in section 2.4.

2.1 Kleinman-Newton method

Several versions of Riccati equations are of practical interest and analyzed in the literature [48, 54, 76, 19]. We focus at first on algebraic Riccati equations (ARE) of the type

$$A^T X + XA - XBB^T X + C^T C = 0 \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{l\times n}$. Other classes of Riccati equations are discussed in section 7.1 and section 7.2.
Above algebraic Riccati equation can be written as a nonlinear system of equations. The goal is to find a symmetric matrix $X \in \mathbb{R}^{n\times n}$ with $\mathcal{F}(X) = 0$, where the map $\mathcal{F} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is defined by

$$\mathcal{F}(X) = A^T X + XA - XBB^T X + C^T C. \tag{2.2}$$

If one applies Newton's method to this system, one has to compute the derivative at X, symmetric, given by

$$\begin{aligned}\mathcal{F}'(X)(Y) &= A^T Y + YA - YBB^T X - XBB^T Y \\ &= (A - BB^T X)^T Y + Y(A - BB^T X) \qquad \forall\ Y \in \mathbb{R}^{n\times n}.\end{aligned}$$

In Newton's method, the next iterate is obtained by solving the Newton system

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) = -\mathcal{F}(X_k) \tag{2.3}$$

which can be also written in an alternative version

$$\mathcal{F}'(X_k)X_{k+1} = \mathcal{F}'(X_k)X_k - \mathcal{F}(X_k). \tag{2.4}$$

For the Riccati equation the computation of a Newton step requires the solution of a Lyapunov equation. Corresponding to (2.4) we obtain

$$X_{k+1}(A - BB^T X_k) + (A - BB^T X_k)^T X_{k+1} = -X_k BB^T X_k - C^T C, \tag{2.5}$$

which is a Lyapunov equation for X_{k+1}.
This method is well understood and analyzed. It does not only exhibit locally a quadratic rate of convergence, but shows also a monotone convergence property, which is not so common for Newton's method and which is due to the quadratic form of $\mathcal{F}$ and the monotonicity of $\mathcal{F}'$. For this to hold, we impose the following definition and assumptions:

Definition 2.1.1. *Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{l\times n}$. A pair (A, BB^T) is called stabilizable if there is a feedback matrix $K \in \mathbb{R}^{n\times n}$ such that $A - BB^T K$ is stable, which means that $A - BB^T K$ has only eigenvalues in the open left half-plane. $(C^T C, A)$ is called detectable if and only if $(A^T, C^T C)$ is stabilizable.*

In the following sections we need the assumption:

Assumption 2.1.2. *(A, BB^T) is stabilizable and (C^TC, A) is detectable.*

Note that by [48, Lemma 4.5.4] the first assumption implies the existence of a matrix X_0 such that $A - BB^TX_0$ is stable.
As a common abbreviation we set

$$A_k := (A - BB^TX_k), \quad k \in \mathbb{N}_0. \tag{2.6}$$

In addition, we introduce a partial ordering on $\mathbb{R}^{n\times n}$ and $A \leq B$ means that the matrix $A - B$ is negative semidefinite.
The next theorem is well known, see e.g. Kleinman [45], Mehrmann [54] or Lancaster and Rodman [48].

Theorem 2.1.3. *Let $X_0 \in \mathbb{R}^{n\times n}$ be symmetric and non-negative definite such that $A - BB^TX_0$ is stable and let Assumption 2.1.2 hold. Then the Newton iterates X_k defined by*

$$X_{k+1}A_k + A_k^TX_{k+1} = -X_kBB^TX_k - C^TC$$

converge to some X_∞ such that $A - BB^TX_\infty$ is stable and it solves the Riccati equation $\mathcal{F}(X_\infty) = 0$. Furthermore the iterates have a monotone convergence behavior

$$0 \leq X_\infty \leq ... \leq X_{k+1} \leq X_k \leq ... \leq X_1$$

and quadratic convergence.

Kleinman was the first who applied Newton's method in the context of Riccati equation, utilizing equation (2.5) for the solution of each Newton step. Therefore we call this version Kleinman-Newton algorithm, which can be presented as follows:

Algorithm 1 Kleinman-Newton method

Require: $X_0 \in \mathbb{R}^{n\times n}$ symmetric and non-negative definite with $A - BB^TX_0$ stable
  for k=0,1,2,... **do**
    Determine a solution X_{k+1} of
    $X_{k+1}(A - BB^TX_k) + (A - BB^TX_k)^TX_{k+1} = -X_kBB^TX_k - C^TC$
  end for

2.2 Inexact Kleinman-Newton methods

In the past decade, a variant of Newton's method has become quite popular in several areas of applications, the so called inexact Newton's method. In this variant, it is no longer necessary to solve the Newton equation exactly for the Newton step, but it is possible to allow for errors in the residual. In particular, this is useful if iterative solvers are used for the solution of the linear Newton equation. We cite a theorem in Kelley [44, p. 99].

Theorem 2.2.1. *Let $F : \mathbb{R}^N \to \mathbb{R}^N$ have a Lipschitz-continuous derivative in a neighborhood of some $x_\infty \in \mathbb{R}^N$ with $F(x_\infty) = 0$ and $F'(x_\infty)$ invertible. Then there exist $\delta > 0$ and $\bar{\eta}$ such that for all $x_0 \in \mathcal{B}(x_\infty, \delta)$ the inexact Newton iterates*

$$x_{k+1} = x_k + s_k$$

where s_k satisfies

$$\|F'(x_k)s_k + F(x_k)\| \leq \eta_k \|F(x_k)\|, \qquad \eta_k \in [0, \bar{\eta}]$$

converge to x_∞. Furthermore we have the following rate estimates:
The rate of convergence is at least linear. If, in addition, $\eta_k \to 0$, then we obtain a superlinear rate and if $\eta_k \leq K_\eta \|F(x_k)\|$ for some $K_\eta > 0$, then we have a quadratic rate of convergence.

Our goal is to analyze, how we can apply the last theorem to the Riccati equation and extend the convergence Theorem 2.1.3 to an inexact Newton's method. This seems to be promising, especially for this application, since in many cases the resulting linear Newton equations are Lyapunov equations which are usually solved iteratively by Smith method or different variants of the ADI method. Some of the results in this section have been already published in [22].
Here we introduce for Riccati equations the inexact Kleinman-Newton method in the context presented in chapter 2.1. Formally, the new iterate is determined by solving

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) + \mathcal{F}(X_k) = R_k \tag{2.7}$$

for X_{k+1}. This can be written more explicitly as a solution of X_{k+1}

$$X_{k+1}A_k + A_k^T X_{k+1} = -X_k BB^T X_k - C^T C + R_k. \tag{2.8}$$

Hence a residual of size R_k is allowed in the k-th Newton step. We summarize the algorithm proposed:

Algorithm 2 Inexact Kleinman-Newton method

Choose $X_0 \in \mathbb{R}^{n\times n}$
for k=0,1,2,... **do**
Determine a solution X_{k+1} of
$X_{k+1}(A - BB^TX_k) + (A - BB^TX_k)^TX_{k+1} = -X_kBB^TX_k - C^TC + R_k$
end for

Before we come to several convergence properties, we recall an existence and uniqueness theorem for Lyapunov equations, which need to be solved at each step of the algorithm.

Theorem 2.2.2. *If $A \in \mathbb{R}^{n\times n}$ is stable, then for each $Z \in \mathbb{R}^{n\times n}$ the Lyapunov equation*

$$A^TY + YA - Z = 0$$

is uniquely solvable and its solution is given by

$$Y = -\int_0^\infty e^{A^Tt}Ze^{At}dt.$$

Proof. For a proof see [48, Theorem 8.5.1]. □

We can formulate local convergence properties for Algorithm 2 by applying a standard theorem about inexact Newton's methods, e.g. Theorem 2.2.1.

Theorem 2.2.3. *Let $X_\infty \in \mathbb{R}^{n\times n}$ be a symmetric solution of (1.1) such that $A - BB^TX_\infty$ is stable. Then there exist $\delta > 0$ and $\bar{\eta} > 0$ such that for all starting values $X_0 \in \mathbb{R}^{n\times n}$ with $\|X_0 - X_\infty\| \le \delta$ the iterates X_k of the inexact Kleinman-Newton Algorithm converge to X_∞, if the residuals R_k satisfy*

$$\|R_k\| \le \eta_k\|\mathcal{F}(X_k)\| = \eta_k\|A^TX_k + X_kA - X_kBB^TX_k + C^TC\|. \tag{2.9}$$

The rate of convergence is linear if $\eta_k \in (0, \bar{\eta}]$, it is superlinear if $\eta_k \to 0$ and quadratic, if $\eta_k \le K_\eta\|\mathcal{F}(X_k)\|$ for some $K_\eta > 0$.

Proof. We apply Theorem 2.2.1 to the equation $\mathcal{F}(X) = A^TX + XA - XBB^TX + C^TC = 0$. This map is differentiable and has a Lipschitz continuous derivative. Since $A - BB^TX_\infty$ is assumed to be stable, $\mathcal{F}'(X_\infty)Y = 0$ implies $Y = 0$ by Theorem 2.2.2, and hence $\mathcal{F}'(X_\infty)$ is an invertible linear map. Since all assumptions in Theorem 2.2.1 hold, the conclusions can be applied and yield the statements in the theorem. □

Above theorem provides stopping criteria for the inexact Kleinman-Newton method (Algorithm 2). The solution of the inner iteration, computed by an iterative Lyapunov solver (chapter 3), can be terminated early, resulting in inexact versions with a linear, superlinear or quadratic rate of convergence.

2.3 Monotonicity results

An interesting fact about the Kleinman-Newton method is that the iterates exhibit monotonicity and a global convergence property, once the initial iterate is such that A_0 is stable. Theorem 1 testifies the monotonicity $X_{k+1} \leq X_k$ for all $k \geq 1$ but the relation between the initial iterate X_0 and X_1 can not be stated beforehand.
These properties are not common for Newton methods and depend on applications of the convexity and monotonicity results. For the inexact version, these identities are perturbed and those results are much harder to obtain. In order to retain the monotonicity of the iterates, we have to impose certain conditions on the residuals.
Let us summarize at first a few monotonicity properties for the Lyapunov operators.

Theorem 2.3.1. *The map $\mathcal{F}$ is concave in following sense:*

$$\mathcal{F}'(X)(Y-X) \geq \mathcal{F}(Y) - \mathcal{F}(X) \qquad \text{for all symmetric } X, Y \in \mathbb{R}^{n\times n} \tag{2.10}$$

Proof. The proof follows directly with the Taylor expansion of the quadratic Riccati equation

$$\mathcal{F}(Y) = \mathcal{F}(X) + \mathcal{F}'(X)(Y-X) + \frac{1}{2}\mathcal{F}''(X)(Y-X, Y-X) \tag{2.11}$$

where the quadratic term

$$\frac{1}{2}\mathcal{F}''(Z)(W,W) = -WBB^TW \leq 0 \tag{2.12}$$

is independent of Z and negative semidefinite. □

Theorem 2.3.2. *Let $A - BB^TX$ be stable. Then*

$$Z = \mathcal{F}'(X)(Y) \Longleftrightarrow Y = -\int_0^\infty e^{(A-BB^TX)^Tt} Z e^{(A-BB^TX)t} dt \tag{2.13}$$

and hence $\mathcal{F}'(X)(Y) \geq 0$ implies $Y \leq 0$.

Proof. We have

$$Z = \mathcal{F}'(X)(Y) = (A - BB^TX)^TY + Y(A - BB^TX).$$

Since $(A - BB^TX)$ is stable, Theorem 2.2.2 yields the result. □

The next theorem shows that we can weaken the condition on the starting point and that the inexact Kleinman-Newton iteration is still well defined.

Theorem 2.3.3. *Let X_k be symmetric and non-negative definite such that $A - BB^T X_k$ is stable and*

$$R_k \leq C^T C \tag{2.14}$$

hold. Then

i) the iterate X_{k+1} of the inexact Kleinman-Newton method is well defined, symmetric and non-negative definite,

ii) and the matrix $A - BB^T X_{k+1}$ is stable.

Proof. The inexact Newton step (2.7) is given by the solution of a Lyapunov equation

$$X_{k+1}A_k + A_k^T X_{k+1} = -X_k BB^T X_k - C^T C + R_k.$$

Since A_k is stable the unique solution X_{k+1} exists and is symmetric by Theorem 2.2.2. Furthermore requirement (2.14) leads to

$$X_{k+1}A_k^T + A_k X_{k+1} \leq 0$$

and Theorem 2.3.2 implies $X_{k+1} \geq 0$. Equation (2.8) is equivalent to

$$\begin{aligned} A_{k+1}^T X_{k+1} + X_{k+1}A_{k+1} = &- C^T C - X_{k+1}BB^T X_{k+1} \\ &- (X_{k+1} - X_k)BB^T(X_{k+1} - X_k) + R_k =: W. \end{aligned} \tag{2.15}$$

We define W as the right side of (2.15).

Let us assume $A_{k+1}x = \lambda x$ for λ with $Re(\lambda) \geq 0$ and $x \neq 0$. Then (2.15) implies

$$(\bar{\lambda} + \lambda)\bar{x}^T X_{k+1} x = \bar{x}^T A_{k+1}^T X_{k+1} x + \bar{x}^T X_{k+1} A_{k+1} x = \bar{x}^T W x.$$

On the one hand, the definition of W combined with requirement (2.14) leads to $W \leq 0$. On the other hand, $X_{k+1} \geq 0$ implies $\bar{x}^T W x = 0$. Using the definition of W and a similar argument as before again, we obtain

$$\bar{x}^T(X_{k+1} - X_k)BB^T(X_{k+1} - X_k)x = 0. \tag{2.16}$$

But $BB^T \geq 0$, so $BB^T(X_{k+1} - X_k)x = 0$ and hence

$$A_{k+1}x = A_k x = \lambda x,$$

contradicting the stability of A_k. Hence A_{k+1} is also stable. □

The requirements on the residuals can be weakened, e.g.

$$R_k \leq C^T C + X_j B B^T X_j \qquad j = k, k+1 \quad \forall k \in \mathbb{N} \tag{2.17}$$

will also admit the previous proof.
In the following theorem we show under which requirements on the residuals R_k, $k \in \mathbb{N}$ the monotonicity of the iterates X_k can be preserved also for the inexact Kleinman-Newton method.

Theorem 2.3.4. *Let Assumption 2.1.2 be satisfied and let X_0, symmetric and positive semi-definite, be such that A_0 is stable. Assume that (2.14) and*

$$0 \leq R_k \leq (X_{k+1} - X_k) B B^T (X_{k+1} - X_k) \tag{2.18}$$

hold for all $k \in \mathbb{N}$. Then the iterates (2.8) satisfy

i) $\lim_{k\to\infty} X_k = X_\infty$ *and* $0 \leq X_\infty \leq ... \leq X_{k+1} \leq X_k \leq ... \leq X_1$,

ii) $(A - BB^T X_\infty)$ *is stable and* X_∞ *is the maximal solution of* $\mathcal{F}(X_\infty) = 0$,

iii) $\|X_{k+1} - X_\infty\| \leq c\|X_k - X_\infty\|^2, k \in \mathbb{N}$.

Proof. Using the definition of an inexact Newton step and (2.11)

$$\begin{aligned} R_k &= \mathcal{F}'(X_k)(X_{k+1} - X_k) + \mathcal{F}(X_k) \\ &= \mathcal{F}(X_{k+1}) + (X_{k+1} - X_k) B B^T (X_{k+1} - X_k). \end{aligned}$$

This can be inserted into the next Newton step

$$\begin{aligned} &\mathcal{F}'(X_{k+1})(X_{k+2} - X_{k+1}) = -\mathcal{F}(X_{k+1}) + R_{k+1} \\ = \;& R_{k+1} - R_k + (X_{k+1} - X_k) B B^T (X_{k+1} - X_k) \geq R_{k+1} \geq 0 \end{aligned}$$

by assumption (2.18). Then from Theorem 2.3.2 we can infer

$$X_{k+2} - X_{k+1} \leq 0, \quad k = 0, 1, 2, ...$$

Therefore $(X_k)_{k\in\mathbb{N}}$ is a monotone sequence of symmetric and non-negative definite matrices and $X_k \geq 0$ due to Theorem 2.3.3. Hence it is convergent to some symmetric and non-negative definite limit matrix

$$\lim_{k\to\infty} X_k = X_\infty.$$

Passing to the limit in (2.7) and (2.18) we deduce that X_∞ satisfies the Riccati equation, $X_\infty \leq X_k$ and $\mathcal{F}(X_\infty) = 0$.
We show that X_∞ is the maximal symmetric solution of the Riccati equation

(1.1), which means $X_\infty \geq X$ for every symmetric solution X of (1.1). For this to hold we assume that X is a symmetric solution of the Riccati equation. Then Theorem 2.3.1 and (2.11) imply

$$\begin{aligned}\mathcal{F}'(X_k)(X - X_k) \geq -\mathcal{F}(X_k) = -\mathcal{F}(X_{k-1}) - \mathcal{F}'(X_{k-1})(X_k - X_{k-1}) - \\ - \frac{1}{2}\mathcal{F}''(X_{k-1})(X_k - X_{k-1}, X_k - X_{k-1}) \geq -R_{k-1}.\end{aligned}$$

Therefore, there exists $Q_k \geq 0$ with

$$\mathcal{F}'(X_k)(X - X_k) = Q_k - R_{k-1}$$

and since A_k is stable Theorem 2.2.2 implies

$$X - X_k = -\int_0^\infty e^{A_k^T t}(Q_k - R_{k-1})e^{A_k t}dt \leq \int_0^\infty e^{A_k^T t} R_{k-1} e^{A_k t}dt. \tag{2.19}$$

Passing to the limits leads to the desired result

$$X - X_\infty \leq 0$$

and X_∞ is the maximal solution. We can deduce from [48, Theorem 9.1.2] that the matrix $A - BB^T X_\infty$ is stable.
To prove the quadratic rate of convergence we use the inexact Newton step

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) + \mathcal{F}(X_k) - R_k = 0$$

and rewrite it using (2.11)

$$\begin{aligned}\mathcal{F}'(X_\infty)(X_{k+1} - X_\infty) &= \mathcal{F}'(X_\infty)(X_{k+1} - X_\infty) - \mathcal{F}(X_{k+1}) + \mathcal{F}(X_\infty) \\ &\quad -(\ \mathcal{F}'(X_k)(X_{k+1} - X_k) - \mathcal{F}(X_{k+1}) + \mathcal{F}(X_k)) + R_k \\ &= (X_{k+1} - X_\infty)BB^T(X_{k+1} - X_\infty) \\ &\quad -(X_{k+1} - X_k)BB^T(X_{k+1} - X_k) + R_k.\end{aligned}$$

Since $A_\infty := (A - BB^T X_\infty)$ is stable, Theorem 2.3.2 shows

$$\begin{aligned}X_{k+1} - X_\infty &= \textstyle\int_0^\infty e^{A_\infty^T t}\{-(X_{k+1} - X_\infty)BB^T(X_{k+1} - X_\infty) \\ &\quad +(X_{k+1} - X_k)BB^T(X_{k+1} - X_k) - R_k\}e^{A_\infty t}\ dt \\ &\leq \textstyle\int_0^\infty e^{A_\infty^T t}((X_{k+1} - X_k)BB^T(X_{k+1} - X_k))e^{A_\infty t}\ dt.\end{aligned} \tag{2.20}$$

Note, that for all symmetric and non-negative matrices $A, B \in \mathbb{R}^{n\times n}$, $A \leq B$ implies $\|A\|_2 \leq \|B\|_2$, due to

$$\lambda_{max}(A) = \max_{\|x\|_2=1} \frac{\bar{x}^T A x}{\bar{x}^T x} = \frac{\bar{x}_*^T A x_*}{\bar{x}_*^T x_*} \leq \frac{\bar{x}_*^T B x_*}{\bar{x}_*^T x_*} \leq \max_{\|x\|_2=1} \frac{\bar{x}^T B x}{\bar{x}^T x} = \lambda_{max}(B).$$

Taking norms in (2.20) we obtain due to the stability of A_∞

$$\begin{aligned} &\|X_{k+1} - X_\infty\|_2 \leq \|X_{k+1} - X_k\|_2^2 \|BB^T\| \int_0^\infty \|e^{A_\infty t}\| \|e^{A_\infty^T t}\| \, dt \\ \leq \;& c\|X_{k+1} - X_k\|_2^2 \end{aligned} \tag{2.21}$$

and using the monotonicity of the iterates

$$0 \leq X_k - X_{k+1} \leq X_k - X_\infty \quad \Rightarrow \quad \|X_k - X_{k+1}\|_2 \leq \|X_k - X_\infty\|_2 \tag{2.22}$$

and therefore

$$\|X_{k+1} - X_\infty\|_2 \leq c\|X_k - X_\infty\|_2^2$$

which implies quadratic convergence in any matrix norm. □

Theorem 2.3.4 together with Theorem 2.3.3 provides a sufficient theoretical background to obtain monotone iterates $X_k \geq X_{k+1},\ k \geq 1$ also for inexact Kleinman-Newton methods. In addition, the initial iterate X_0 does not have to be close to the solution, only the stability of the matrix $A - BB^T X_0$ plays a crucial role. In summary, we were able to extend the well-known convergence results of Kleinman (Theorem 2.1.3) by introducing additional requirements on the residuals of the Newton steps.
We impose several requirements on the residuals in Theorem 2.3.3 and Theorem 2.3.4. Some of them restrict the size of R_k in dependence on the step, see (2.18) and (2.14), others assume the non-negative definiteness, i.e. $R_k \geq 0,\ k \in \mathbb{N}$.
The latter assumption is a condition, which the iterative Lyapunov solver has to satisfy, like Smith method or variants of the ADI method. Since this condition is an essential part of our theory, we consider the question whether these iterative solvers provide non-negative definite residuals in section 3.5.
The other assumption on the size of the residuals e.g. in (2.18) depends on the quantity X_{k+1}, which has to be computed by the iterative procedure. However, the inequality involved can be tested as the iteration for X_{k+1} progresses. Of course, the verification of a requirement such as $R_k \leq C^T C$ involves a significant numerical effort.

2.4 Robustness

Let us note that a second implementation of Newton's method for the solution of the algebraic Riccati equation (1.1) is presented in the literature, e.g. [3], [55]. In practice, this version shows some instabilities which have not been yet explained in a satisfactory theoretical manners. The convergence theory for inexact Newton's methods enables us to understand and analyze these phenomena, see also

[41] for a discussion on this topic.
Banks and Ito [3] introduced an alternative implementation, where the Newton step is computed by a Lyapunov equation for the increment $X_{k+1} - X_k$ in the following way

$$\begin{aligned}(X_{k+1} - X_k)(A - BB^T X_k) + (A - BB^T X_k)^T (X_{k+1} - X_k) \\ = (X_k - X_{k-1})BB^T(X_k - X_{k-1}).\end{aligned} \tag{2.23}$$

Against one espectations this Lyapunov equation does not represent the matrix notation of the Newton step

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) = -\mathcal{F}(X_k). \tag{2.24}$$

In order to establish the iteration (2.23) one significant modification is necessary. This variation is based on an identity due to the quadratic nature of the algebraic Riccati equation:

$$\mathcal{F}(Y) = \mathcal{F}(X) + \mathcal{F}'(X)(Y - X) + \frac{1}{2}\mathcal{F}''(X)(Y - X, Y - X) \tag{2.25}$$

with the quadratic term

$$\frac{1}{2}\mathcal{F}''(Z)(W, W) = -WBB^T W.$$

If we use (2.25) for Newton's method, we obtain

$$\begin{aligned}\mathcal{F}'(X_k)(X_{k+1} - X_k) = -\mathcal{F}(X_k) \\ = -\mathcal{F}(X_{k-1}) - \mathcal{F}'(X_{k-1})(X_k - X_{k-1}) - \tfrac{1}{2}\mathcal{F}''(X_{k-1})(X_k - X_{k-1}, X_k - X_{k-1}) \\ = -\tfrac{1}{2}\mathcal{F}''(X_{k-1})(X_k - X_{k-1}, X_k - X_{k-1})\end{aligned} \tag{2.26}$$

where the Newton step for the previous iterate X_k was exploited for the last equality. Now we can identify the right hand side of the alternative version (2.23) with the matrix notation of

$$-\frac{1}{2}\mathcal{F}''(X_{k-1})(X_k - X_{k-1}, X_k - X_{k-1}).$$

Hence the Newton step can be alternatively computed by a Lyapunov equation for its increment $X_{k+1} - X_k$ in the following way

$$\begin{aligned}(X_{k+1} - X_k)(A - BB^T X_k) + (A - BB^T X_k)^T (X_{k+1} - X_k) \\ = (X_k - X_{k-1})BB^T(X_k - X_{k-1}),\end{aligned}$$

in contrast to the Newton step (2.5) of the standard Kleinman-Newton method (Algorithm 1)

$$X_{k+1}(A - BB^T X_k) + (A - BB^T X_k)^T X_{k+1} = -X_k BB^T X_k - C^T C.$$

Note that the inhomogeneous terms in the Lyapunov equations for both variants of Newton's method differ quite substantially. In (2.5) the right hand side is

$$-(X_k BB^T X_k + C^T C)$$

whereas in (2.23) we have

$$(X_k - X_{k-1})BB^T(X_k - X_{k-1}),$$

which e.g. does not depend on C.
The authors of [3] pointed out that the second version exhibit some advantages compared to the standard implementation. Since B is often a low rank matrix, the right side

$$(X_k - X_{k-1})BB^T(X_k - X_{k-1}),$$

is of low numerical rank, independent of the matrix C. Therefore an efficient low-rank Cholesky ADI method (Algorithm 14) can always be applied to these Lyapunov equations. Another advantage is the possibility to develop a feedback gain algorithm with the help of this second implementation, see chapter 5 for details.
This second implementation can be outlined as follows:

Algorithm 3 Kleinman-Newton method (Version 2)

Require: $X_0, X_1 \in \mathbb{R}^{n\times n}$
  for k=1,2,... **do**
    Determine a solution $\triangle X_k$ of
    $\triangle X_k(A - BB^T X_k) + (A - BB^T X_k)^T \triangle X_k = (X_k - X_{k-1})BB^T(X_k - X_{k-1})$
    Set $X_{k+1} = X_k + \triangle X_k$
  end for

Note, for the initialization of Algorithm 3 two stable initial iterates X_0 and X_1 are necessary, whereas the standard implementation only requires X_0. Usually the second initial iterate X_1 is determined via one step of the classical Kleinman-Newton algorithm (Algorithm 2).
We can show the precise statement for the equivalence of both versions in case of exact Newton's method with the following Lemma:

Lemma 2.4.1. *If a sequence X_k satisfies (2.5), then it also fulfills (2.23). If, conversely, a sequence X_k satisfies (2.23), then it also fulfills (2.5), provided the starting points X_0, X_1 satisfy (2.5) for $k = 0$.*

Proof. The first conclusion was shown in (2.26) since the iterates X_k of the standard implementation, defined in (2.5), obviously satisfy the Newton equation. For the reverse to hold we use (2.26) and (2.25) and obtain

$$\begin{aligned}\mathcal{F}'(X_{k+1})(X_{k+2} - X_{k+1}) &= -\tfrac{1}{2}\mathcal{F}''(X_k)(X_{k+1} - X_k, X_{k+1} - X_k) \\ &= -\mathcal{F}(X_{k+1}) + \mathcal{F}(X_k) + \mathcal{F}'(X_k)(X_{k+1} - X_k)\end{aligned}$$

and hence

$$\mathcal{F}(X_{k+1}) + \mathcal{F}'(X_{k+1})(X_{k+2} - X_{k+1}) = \mathcal{F}(X_k) + \mathcal{F}'(X_k)(X_{k+1} - X_k)$$

for all $k \geq 0$. Since it is assumed that for the starting iterates

$$\mathcal{F}(X_0) + \mathcal{F}'(X_0)(X_1 - X_0) = 0, \tag{2.27}$$

the X_k also satisfy the Newton equation and therefore (2.5) holds. □

While Lemma 2.4.1 proves that both methods are identical for the exact case if the first iterates are chosen appropriately, this does not hold anymore in the inexact case.
Let us first state the inexact variant:

Algorithm 4 Inexact Kleinman-Newton method (Version 2)

Choose X_0, X_1 satisfying (2.27)
for k=1,2,... **do**
  Determine a solution $\triangle X_k$
  which solves the Lyapunov equation up to a residual $\tilde{R}_k$
  $\triangle X_k(A - BB^T X_k) + (A - BB^T X_k)^T \triangle X_k = (X_k - X_{k-1})BB^T(X_k - X_{k-1}) + \tilde{R}_k$
  Set $X_{k+1} = X_k + \triangle X_k$
end for

In order to state a convergence result for this implementation, we have to reformulate the steps in Algorithm 4.
Using the formulation with $\mathcal{F}$ and (2.26), the Newton step can be rewritten as:

$$\begin{aligned}\mathcal{F}'(X_k)(X_{k+1} - X_k) &= -\tfrac{1}{2}\mathcal{F}''(X_{k-1})(X_k - X_{k-1}, X_k - X_{k-1}) + \tilde{R}_k \\ &= \mathcal{F}(X_{k-1}) + \mathcal{F}'(X_{k-1})(X_k - X_{k-1}) - \mathcal{F}(X_k) + \tilde{R}_k\end{aligned}$$

or equivalently

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) + \mathcal{F}(X_k) = \mathcal{F}'(X_{k-1})(X_k - X_{k-1}) + \mathcal{F}(X_{k-1}) + \tilde{R}_k.$$

Using this recursively yields

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) + \mathcal{F}(X_k) = \sum_{i=1}^{k} \tilde{R}_i.$$

According to the convergence theory of inexact Newton's methods (e.g. Theorem 2.2.1) one would have to bound $R_k = \sum_{i=1}^{k} \tilde{R}_i$ (if X_1 is computed by an exact Newton step), which seems to be a rather strong assumption.
It is important to notice that the residuals are accumulatory. The second version therefore exhibits a kind of instability and is no longer self-correcting. This implies that it is nearly impossible to develop an inexact version based on the second implementation.
The inapplicability of the second implementation raises some difficulties in the context of feedback gain algorithms because several feedback gain algorithms are built with this formulation, e.g. [3], [55]. We present some ideas to circumvent these problems in chapter 5.

Chapter 3

Iterative methods for Lyapunov equations

As shown in chapter 2, each step of the Kleinman-Newton method is equivalent to the solution of a corresponding Lyapunov equation. Recall that at Newton iteration step k the following equation needs to be solved for $X = X_{k+1}$

$$XA_k + A_k^T X + S_k = 0 \tag{3.1}$$

with a stable matrix A_k

$$A_k = A - BB^T X_k \in \mathbb{R}^{n\times n}, \quad S_k = X_k BB^T X_k + C^T C.$$

There is a sizeable amount of literature on how to solve Lyapunov equations with direct solvers and iterative methods.
In the inexact context we do not address direct Lyapunov solvers as presented e.g. in Laub [49], Roberts [62] or Grasedyck [29].
We concentrate on several iterative methods, which are especially important for large scale Lyapunov equations. The inexact Newton's method allows for early termination of these iterations, because the convergence criterion is not so stringent far away from the solution.
In the next section we review Smith method. Chapter 3.2 summarizes the ADI method for Riccati equations. Different modifications of Smith method are presented in the literature and outlined in section 3.3. Smith method and the ADI method are the basis of so called low-rank algorithms, which are nowadays state-of-the-art and are presented in chapter 3.4.
For the monotonicity preserving convergence theory, presented in section 2.3, the non-negative definiteness of the residuals, provided by an iterate of the Lyapunov solver, is essential. The question, whether the above mentioned iterative solvers fit to this assumption, is answered in the main section 3.5 of this chapter.

3.1 Smith method

One of the first iterative methods to solve Lyapunov equations was developed by Smith [70]. It is based on the fact that the solution of a Lyapunov equation is equivalent to the solution of a corresponding Stein's equation. Several modifications and generalizations of Smith method are presented in the literature and reviewed in chapter 3.3.
We slightly modify the Lyapunov equation of the Newton step (3.1) by introducing a factorization of the right side S_k using a matrix $D_k := \begin{bmatrix} B^T X_k \\ C \end{bmatrix} \in \mathbb{R}^{(l+m)\times n}$. Now we are able to rewrite $S_k = D_k^T D_k$, which will be useful in the next sections.
In the following we motivate Smith method [70] to solve

$$X A_k + A_k^T X + D_k^T D_k = 0 \tag{3.2}$$

for $X = X_{k+1}$. Note that this equation is equivalent to a Stein's equation:

Lemma 3.1.1. *Given any $\mu \in \mathbb{R}^- := \{x \in \mathbb{R} |\ x < 0\}$, then a solution X of the Lyapunov equation (3.2) is also a solution of Stein's equation and vice versa. Stein's equation is given by*

$$X = A_{k,\mu}^T X A_{k,\mu} + S_{k,\mu} \tag{3.3}$$

with

$$A_{k,\mu} = (A_k - \mu I)(A_k + \mu I)^{-1}, \quad S_{k,\mu} = -2\mu(A_k^T + \mu I)^{-1} D_k^T D_k (A_k + \mu I)^{-1}.$$

Proof. Note that (3.2) is equivalent to

$$(A_k^T + \mu I)X(A_k + \mu I) - (A_k^T - \mu I)X(A_k - \mu I) = -2\mu D_k^T D_k$$

and from this (3.3) follows. Since A_k is assumed to be stable, all eigenvalues of $A_k + \mu I$ have negative real parts for any $\mu \in \mathbb{R}^-$, which secures the existence of $(A_k + \mu I)^{-1}$.

□

The resulting algorithm to solve equation (3.3) and therefore (3.2) is a fixpoint iteration for (3.3) and can be presented as follows:

Algorithm 5 Smith method

Require: $X_{k+1,0} = 0 \in \mathbb{R}^{n\times n}$, shift parameter $\mu \in \mathbb{R}^-$
  Define: $A_{k,\mu} = (A_k - \mu I)(A_k + \mu I)^{-1}$, $S_{k,\mu} = -2\mu(A_k^T + \mu I)^{-1} D_k^T D_k (A_k + \mu I)^{-1}$
  for i=1,2,... **do**
    $X_{k+1,i} = A_{k,\mu}^T X_{k+1,i-1} A_{k,\mu} + S_{k,\mu}$
  end for

Note, that the iterates to determine X_{k+1} are called $X_{k+1,i},\ i \in \mathbb{N}_0$.
The convergence of this method is stated in the next theorem.

Theorem 3.1.2. *Let $\mu \in \mathbb{R}^-$ then Stein's equation $X = A_{k,\mu}^T X A_{k,\mu} + S_{k,\mu}$ has a solution*

$$X = \sum_{j=0}^{\infty} (A_{k,\mu}^T)^j S_{k,\mu} A_{k,\mu}^j = \lim_{i\to\infty} X_{k+1,i}. \tag{3.4}$$

If $S_{k,\mu}$ is symmetric, X is also symmetric. If $S_{k,\mu} \geq 0$, then $X \geq S_{k,\mu}$.

Proof. This can be proved by showing that there exists X such that iterates

$$X_{k+1,i} := \sum_{j=0}^{i-1} (A_{k,\mu}^T)^j S_{k,\mu} A_{k,\mu}^j \underset{i\to\infty}{\longrightarrow} X \tag{3.5}$$

due to $\rho(A_{k,\mu}) = \max\limits_{\lambda\in\sigma(A_k)} |\frac{\lambda-\mu}{\lambda+\mu}| < 1$ for every $\mu \in \mathbb{R}^-$. X solves Stein's equation because

$$\begin{aligned} A_{k,\mu}^T X_{k+1,i} A_{k,\mu} &= \sum_{j=0}^{i-1} A_{k,\mu}^T (A_{k,\mu}^T)^j S_{k,\mu} A_{k,\mu}^j A_{k,\mu} \\ &= \sum_{j=1}^{i} (A_{k,\mu}^T)^j S_{k,\mu} A_{k,\mu}^j = X_{k+1,i+1} - S_{k,\mu}. \end{aligned} \tag{3.6}$$

Taking the limit for $i \to \infty$ yields the result. The definition of $X_{k+1,i}$ together with the non-negative definiteness of $S_{k,\mu}$ leads to the conclusion $X \geq S_{k,\mu}$. □

The convergence behavior of the algorithm proposed depends on the choice of the underlying shift parameter. There exits a huge amount of parameter selection methods because this problem is also important for the ADI method presented in chapter 3.2. An extensive overview of existing parameter selection methods can be found in [8] or [67] and the references therein.

3.2 ADI method

Peaceman and Rachford [57] introduced the ADI (Alternating Direction Implicit) method to solve a special kind of linear systems arising from the discretization of elliptic boundary value problems. It is also possible to apply this method to the solution of Lyapunov equations, which has been shown by Wachspress [52]. There exist two versions of the ADI method. At first we review a formulation

which is build with two steps. In order to compare the ADI method with Smith method, we reformulate the ADI method into an one step version.
The ADI algorithm to solve Lyapunov equation (3.2), as presented in [59], can be outlined:

Algorithm 6 ADI method (two steps)

Choose $X_{k+1,0} \in \mathbb{R}^{n\times n}$, set of shift parameter $\mu_i \in \mathbb{R}^-, i \in \mathbb{N}$
for i=1,2,... **do**
$(A_k^T + \mu_i I)X_{k+1,i-\frac{1}{2}} = -D_k^T D_k - X_{k+1,i-1}(A_k - \mu_i I)$
$(A_k^T + \mu_i I)X_{k+1,i}^T = -D_k^T D_k - X_{k+1,i-\frac{1}{2}}^T(A_k - \mu_i I)$
end for

This method is closely related to Smith method, because it is possible to combine both steps of Algorithm 6.

Lemma 3.2.1. *The two step iteration loop of the ADI method (Algorithm 6)*

$$(A_k^T + \mu_i I)X_{k+1,i-\frac{1}{2}} = -D_k^T D_k - X_{k+1,i-1}(A_k - \mu_i I)$$

$$(A_k^T + \mu_i I)X_{k+1,i}^T = -D_k^T D_k - X_{k+1,i-\frac{1}{2}}^T(A_k - \mu_i I)$$

can be restated in an equivalent one step version

$$X_{k+1,i} = A_{k,\mu_i}^T X_{k+1,i-1} A_{k,\mu_i} + S_{k,\mu_i}.$$

Proof. In order to develop an one step version, we have to reformulate the first step of Algorithm 6

$$X_{k+1,i-\frac{1}{2}} = (A_k^T + \mu_i I)^{-1}(-D_k^T D_k - X_{k+1,i-1}(A_k - \mu_i I))$$

and insert $X_{k+1,i-\frac{1}{2}}$ into the next step:

$$\begin{aligned}
\Longrightarrow \quad X_{k+1,i}^T = &-(A_k^T + \mu_i I)^{-1} D_k^T D_k + \\
&(A_k^T + \mu_i I)^{-1} D_k^T D_k (A_k + \mu_i I)^{-1}(A_k - \mu_i I) + \\
&(A_k^T + \mu_i I)^{-1}(A_k^T - \mu_i I)X_{k+1,i-1}^T(A_k + \mu_i I)^{-1}(A_k - \mu_i I) \\
\Longleftrightarrow \quad X_{k+1,i}^T = &(A_k^T + \mu_i I)^{-1} D_k^T D_k(-I + (A_k + \mu_i I)^{-1}(A_k - \mu_i I)) + \\
&(A_k^T + \mu_i I)^{-1}(A_k^T - \mu_i I)X_{k+1,i-1}^T(A_k + \mu_i I)^{-1}(A_k - \mu_i I).
\end{aligned}$$

Transposing and substituting $I = (A_k + \mu_i I)^{-1}(A_k + \mu_i I)$ leads to

$$\begin{aligned}
\Longleftrightarrow \quad X_{k+1,i} = &-2\mu_i (A_k^T + \mu_i I)^{-1} D_k^T D_k (A_k + \mu_i I)^{-1} + \\
&(A_k^T - \mu_i I)(A_k^T + \mu_i I)^{-1} X_{k+1,i-1} \underbrace{(A_k - \mu_i I)(A_k + \mu_i I)^{-1}}_{=:A_{k,\mu_i}}.
\end{aligned}$$

Finally we have to realize that

$$(A_k^T - \mu_i I)(A_k^T + \mu_i I)^{-1} \stackrel{!}{=} A_{k,\mu_i}^T$$

which can be verified due to

$$\begin{aligned} & (A_k^T - \mu_i I)(A_k^T + \mu_i I)^{-1} \stackrel{!}{=} A_{k,\mu_i}^T = (A_k^T + \mu_i I)^{-1}(A_k^T - \mu_i I) \\ \iff \quad & (A_k^T + \mu_i I)(A_k^T - \mu_i I) = (A_k^T - \mu_i I)(A_k^T + \mu_i I) \\ \iff \quad & 0 = 0. \end{aligned}$$

Now we are able to rewrite the ADI method in a one step version

$$\begin{gathered} X_{k+1,i} = A_{k,\mu_i}^T X_{k+1,i-1} A_{k,\mu_i} - 2\mu_i (A_k^T + \mu_i I)^{-1} D_k^T D_k (A_k + \mu_i I)^{-1} \\ \iff \\ X_{k+1,i} = A_{k,\mu_i}^T X_{k+1,i-1} A_{k,\mu_i} + S_{k,\mu_i} \end{gathered}$$

with

$$A_{k,\mu_i} = (A_k - \mu_i I)(A_k + \mu_i I)^{-1}, \quad S_{k,\mu_i} = -2\mu_i (A_k^T + \mu_i I)^{-1} D_k^T D_k (A_k + \mu_i I)^{-1}.$$

□

This one step version can be stated:

Algorithm 7 ADI method (one step)

Choose $X_{k+1,0} \in \mathbb{R}^{n \times n}$, set of shift parameter $\mu_i \in \mathbb{R}^-,\ i \in \mathbb{N}$
for i=1,2,... **do**
$A_{k,\mu_i} = (A_k - \mu_i I)(A_k + \mu_i I)^{-1}, \quad S_{k,\mu_i} = -2\mu_i (A_k^T + \mu_i I)^{-1} D_k^T D_k (A_k + \mu_i I)^{-1}$
$X_{k+1,i} = A_{k,\mu_i}^T X_{k+1,i-1} A_{k,\mu_i} + S_{k,\mu_i}$
end for

The relation to Smith method, presented in Algorithm 5, becomes now clear. Both iteration loops are identical except of the used shift parameters. ADI method utilizes different shift parameters $\mu_i, i \in \mathbb{N}$ for each step in contrast to Smith method where the same parameter μ is used for all iterations.
Obviously, the convergence of the proposed algorithm is crucially dependent on the applied set of shift parameters. If the parameter are chosen appropriately the convergence rate of the ADI method will be superlinear. The authors of [8] and [67] present an interesting discussion on parameter selection methods.
These parameter selection procedures have an enormous numerical effort in common. Here one should always consider the relation to the computing time

of the needed ADI iterations and the improvements gained due to better shift parameters.
In order to reduce this effort it is often common practice to determine only a finite set of shift parameter $\tilde{\mu_1}, ..., \tilde{\mu_s}$, which are used in a cyclic manner, i.e. $\mu_{i+\lfloor \frac{j}{s} \rfloor s} = \tilde{\mu_i}$ for $i = j \ mod \ s$ and $j \in \mathbb{N}$, $\tilde{\mu_0} := \tilde{\mu_s}$. This version of the ADI method is called cyclic ADI method:

Algorithm 8 Cyclic ADI method

Choose $X_{k+1,0} \in \mathbb{R}^{n \times n}$, finite set of shift parameter $\tilde{\mu_1}, ..., \tilde{\mu_s} \in \mathbb{R}^-$
Define $\mu_{i+js} = \tilde{\mu_i}$ for $i = 1, ..., s$ and $j = 0, 1, ...$
for i=1,2,... **do**
$A_{k,\mu_i} = (A_k - \mu_i I)(A_k + \mu_i I)^{-1}, \quad S_{k,\mu_i} = -2\mu_i (A_k^T + \mu_i I)^{-1} D_k^T D_k (A_k + \mu_i I)^{-1}$
$X_{k+1,i} = A_{k,\mu_i}^T X_{k+1,i-1} A_{k,\mu_i} + S_{k,\mu_i}$
end for

An implementation of the cyclic ADI method based on two steps can be easily obtained analogous to the one step variant, mentioned above.

3.3 Modifications of Smith method

Several modifications of Smith method have been introduced. We present briefly two established methods, the squared Smith method [70] and the cyclic Smith method [59]. A third modification [3], which is not well known, will be discussed in detail.
One Algorithm, the squared Smith method, has been developed by Smith himself, for details see [70], [60]. In this method only a subsequence $\{X_{k+1,2^i}\}_{i=0}^{\infty}$ of the original Smith iterates $X_{k+1,i}$, $i \in \mathbb{N}$ is determined:

Algorithm 9 Squared Smith method

Require: $X_{k+1,0} = 0 \in \mathbb{R}^{n \times n}$, shift parameter $\mu \in \mathbb{R}^-$
Define: $A_{k,\mu} = (A_k - \mu I)(A_k + \mu I)^{-1}$,$S_{k,\mu} = -2\mu (A_k^T + \mu I)^{-1} D_k^T D_k (A_k + \mu I)^{-1}$
$X_{k+1,2^0} = S_{k,\mu}$
for i=0,1,... **do**
$X_{k+1,2^{i+1}} = X_{k+1,2^i} + (A_{k,\mu}^{2^i})^T X_{k+1,2^i} A_{k,\mu}^{2^i}$
end for

The iteration loop of the squared version is based on the structure of the iterates of Smith method (3.5). For an efficient implementation an economic update

scheme for the matrices $A_{k,\mu}^{2^i}$ is needed, e.g. [1]. Remarks on the computational costs and convergence results for this method can be found in [70] and [60].
Another generalization of the Smith method is presented in [60]. Penzl compared the performance of cyclic ADI methods utilizing sets of shift parameter with varying number of shift parameter. The performance of the cyclic ADI method improves as the number of shift parameters increases. However, the improvement of an additional shift parameter diminishes with increasing number of shift parameter. In order to take benefit of this observation Penzl developed the cyclic Smith method, which is closely related to the cyclic case of the ADI method, presented in chapter 3.2, where a finite set of shift parameter $\mu_1, ..., \mu_s$ is used in a cyclic manner. To initialize the cyclic Smith method, it is necessary to compute the s-th iterate $X_{k+1,s}^{ADI}$ of the ADI method with the given set of shift parameters $\mu_1, ..., \mu_s$. The proposed algorithm computes only a subsequence of the cyclic ADI iterates and can be outlined:

Algorithm 10 Cyclic Smith method

Require: $X_{k+1,0} = 0 \in \mathbb{R}^{n\times n}$, set of shift parameter $\mu_1, ..., \mu_s \in \mathbb{R}^-$
Require: $X_{k+1,s}^{ADI}$ according to Algorithm 6 with above parameter
Define: $S_{k,\mu_1,...,\mu_s} = \prod_{j=1}^{s} (A_k - \mu_j I)(A_k + \mu_j I)^{-1}$
for i=0,1,... **do**
$X_{k+1,(i+1)s} = X_{k+1,s}^{ADI} + S_{k,\mu_1,...,\mu_s}^T X_{k+1,is} S_{k,\mu_1,...,\mu_s}$
end for

The sequence $\{X_{k+1,is}\}_{i=0}^{\infty}$ is a subsequence of the original cyclic ADI iterates, assuming the use of cyclic shift parameter $\tilde{\mu_1}, ..., \tilde{\mu_s}$, for a proof see [59].
The third method has not been introduced as a modification of Smith method and is therefore not well known. In order to develop a feedback gain algorithm, Banks and Ito [3] rewrote Smith method as an equivalent algorithm, but this new algorithm exhibit some numerical benefits compared to the original version.
In the following we review their modification of Smith method, which result in a factored form of the Smith method.
Remember that at Newton iteration step k the Lyapunov equation

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) + \mathcal{F}(X_k) = 0,$$

needs to be solved, or as in (2.8) we solve for $X = X_{k+1}$

$$X A_k + A_k^T X + D_k^T D_k = 0 \tag{3.7}$$

with a stable A_k

$$A_k = A - BB^T X_k, \quad D_k := [B^T X_k \;\; C] \in \mathbb{R}^{(l+m)\times n}. \tag{3.8}$$

We now follow the ideas given by [3] to develop a modification of the Smith method:
Subtracting two iterates of the original Smith algorithm (Algorithm 5) leads to

$$X_{k+1,i+1} - X_{k+1,i} = A_{k,\mu}^T (X_{k+1,i} - X_{k+1,i-1}) A_{k,\mu} \qquad \forall\ i \geq 1. \tag{3.9}$$

If we can find a factorization of the increment $X_{k+1,i} - X_{k+1,i-1} = M_{k+1,i}^T M_{k+1,i}$, then we will be able to rewrite above equation

$$X_{k+1,i+1} - X_{k+1,i} = A_{k,\mu}^T M_{k+1,i}^T M_{k+1,i} A_{k,\mu} = (M_{k+1,i} A_{k,\mu})^T M_{k+1,i} A_{k,\mu}. \tag{3.10}$$

The choice of the zero matrix as a starting point $X_{k+1,0} = 0 \in \mathbb{R}^{n \times n}$ guarantees, in combination with the definition of $S_{k,\mu}$ in Algorithm 5, the existence of such a factorization for the first step :

$$\begin{aligned} X_{k+1,1} - X_{k+1,0} &= -2\mu M_{k+1,1}^T M_{k+1,1} \qquad (3.11)\\ with \quad M_{k+1,1} &:= D_k (A_k + \mu I)^{-1} \quad \in \mathbb{R}^{(l+m)\times n} \end{aligned}$$

Now it is possible to update the factorization $M_{k+1,i} A_{k,\mu}, i \in \mathbb{N}$ of the difference $X_{k+1,i+1} - X_{k+1,i}$ instead of working with the original iterates $X_{k+1,i}, i \in \mathbb{N}$. We develop a new iteration loop for Algorithm 5 based on (3.9) and (3.10)

$$M_{k+1,i+1} = M_{k+1,i} A_{k,\mu} \qquad \in \mathbb{R}^{(l+m)\times n} \tag{3.12}$$

$$X_{k+1,i+1} = X_{k+1,i} - 2\mu M_{k+1,i+1}^T M_{k+1,i+1}, \tag{3.13}$$

which is only a reformulation of the original Smith method:

Algorithm 11 Factored Smith method

Require: stable matrix A_k, D_k according to (3.8) and shift parameter $\mu \in \mathbb{R}^-$
  Define: $A_{k,\mu} = (A_k - \mu I)(A_k + \mu I)^{-1}$, $M_{k+1,1} = D_k (A_k + \mu I)^{-1}$
Ensure: $X_{k+1,0} = 0$, $X_{k+1,1} = -2\mu M_{k+1,1}^T M_{k+1,1}$
  for i=1,2,... **do**
    $M_{k+1,i+1} = M_{k+1,i} A_{k,\mu}$
    $X_{k+1,i+1} = X_{k+1,i} - 2\mu M_{k+1,i+1}^T M_{k+1,i+1}$
  end for

But this factored version of Smith method exhibit some numerical benefits, e.g. is the number of flops with $O(n^2(l+m))$ usually better than the flop count $O(n^3)$ of the original Smith method. Remember that m and l are usually much smaller than n.

Comparing this algorithm with the low-rank Smith method (Algorithm 12), presented in chapter 3.4, leads to a surprising conclusion. Banks and Ito's modification [3] can be seen as an intermediate step in developing the low-rank algorithms. Introducing a factorization of the iterates $X_{k+1,i} = L_{k+1,i}^T L_{k+1,i}$ would yield a low-rank version of Smith method, related to Algorithm 12.
Due to above modification it is possible to develop a feedback gain algorithm. This has been also considered by Banks and Ito [3]. But their approach was based on the second implementation of the Kleinman-Newton method and is therefore not suitable in the inexact context, as shown in chapter 2.4. Nevertheless the modified Smith method developed there can be adapted for the standard implementation too, which is presented in section 5.

3.4 Low rank algorithms

Especially for large scale systems it is necessary to take storage requirements into account. Note that Smith method and the ADI method need storage requirement of size $O(n^2)$ to save the dense actual iterate $X_{k+1,i}$. This can be seen as a disadvantage of both methods, which also hold for their modifications, presented in chapter 3.3.
In order to reduce this drawback, several so-called low-rank algorithms have been considered. Penzl [59] introduced low-rank versions of Smith method, cyclic Smith and ADI method. Independently the low-rank ADI method has been developed by Li, Wang and White [50] and improved by Li and White [51]. An interesting discussion on these algorithms can be found in [7] and an extension of the low-rank Smith Method is given in [31].
In this section we present the key idea of low-rank algorithms. Additionally we introduce two versions of low-rank ADI methods because of their practical importance.
In order to reduce the storage requirements one does no longer work with the original iterates $X_{k+1,i},\ i \in \mathbb{N}_0$. Instead of updating $X_{k+1,i} \in \mathbb{R}^{n\times n}$ it is possible to update a factorization of the type

$$X_{k+1,i} = L_{k+1,i} L_{k+1,i}^T$$

with a matrix $L_{k+1,i}$, which dimension is usually small compared to the dimension of $X_{k+1,i}$. Note, that is no longer necessary to store the iterate $X_{k+1,i},\ i \in \mathbb{N}_0$ only the low-rank factors are needed for further computations. Since all iterates are non-negative definite, assuming a non-negative definite initial iterate $X_{k+1,0}$, the existence of such a factorization is guaranteed. The structure of the factorization is dependent on the considered algorithm.

At first we derive a low-rank version of Smith method (Algorithm 5), as presented in section 3.1. The initial iterate is set $L_{k+1,0} = 0 \in \mathbb{R}^{n\times 0}$, such that $L_{k+1,0}L_{k+1,0}^T$ theoretically represents the initial iterate of Smith method $X_{k+1,0} = 0 \in \mathbb{R}^{n\times n}$. Note, that the definition of $S_{k,\mu}$ enables us to rewrite the Smith iteration (Algorithm 5) in a factored form:

$$\begin{aligned} X_{k+1,i} &= A_{k,\mu}^T X_{k+1,i-1} A_{k,\mu} + S_{k,\mu} \quad \Longleftrightarrow \\ L_{k+1,i}L_{k+1,i}^T &= A_{k,\mu}^T L_{k+1,i-1}L_{k+1,i-1}^T A_{k,\mu} \underbrace{-2\mu(A_k^T + \mu I)^{-1}D_k^T D_k (A_k + \mu I)^{-1}}_{S_{k,\mu}} \end{aligned}$$

Choosing $L_{k+1,0} = 0 \in \mathbb{R}^{n\times 0}$ leads to $L_{k+1,1} = \sqrt{-2\mu}(A_k^T + \mu I)^{-1}D_k^T$ and the factors $L_{k+1,i},\ i > 1$ can now be easily derived from

Algorithm 12 Low-rank Smith method

Require: shift parameter $\mu \in \mathbb{R}^-$
  Define: $A_{k,\mu} = (A_k - \mu I)(A_k + \mu I)^{-1}$, $D_k := [C^T \quad X_k B]^T \in \mathbb{R}^{(l+m)\times n}$
  $L_{k+1,1} = \sqrt{-2\mu}(A_k^T + \mu I)^{-1}D_k^T$
  for i=2,3... **do**
    $L_{k+1,i} = [A_{k,\mu}^T L_{k+1,i-1} \quad , \quad L_{k+1,1}]$
  end for

Note, that the dimension of the iterates increase as the iteration for $L_{k+1,i}$ progresses. $L_{k+1,i}$ is of dimension $n \times i(l+m)$. Recall that $m << n$ and $l << n$, and therefore such kind of algorithms are called low-rank algorithms.
Gugercin et al [31] utilize singular value decomposition to store the factored iterates and update the singular value decomposition for each step, instead of recomputing it. In contrast to the original low-rank Smith algorithm, the dimension of the iterates does no longer necessary increase with each step.
An analogous ansatz as in the Smith case leads to a low-rank version of the ADI method. Corresponding to $X_{k+1,0} = 0 \in \mathbb{R}^{n\times n}$ we denote $L_{k+1,0} = 0 \in \mathbb{R}^{n\times 0}$. Remember the one step ADI iteration loop (Algorithm 7), as presented in section 3.2, written in a factored form:

$$\begin{aligned} X_{k+1,i} &= A_{k,\mu_i}^T X_{k+1,i-1} A_{k,\mu_i} + S_{k,\mu_i} \quad \Longrightarrow \\ L_{k+1,i}L_{k+1,i}^T &= A_{k,\mu_i}^T L_{k+1,i-1}L_{k+1,i-1}^T A_{k,\mu_i} \underbrace{-2\mu_i(A_k^T + \mu_i I)^{-1}D_k^T D_k (A_k + \mu_i I)^{-1}}_{S_{k,\mu_i}} \end{aligned}$$

Due to the choice of the initial iterate, we obtain $L_{k+1,1} = \sqrt{-2\mu_1}(A_k^T + \mu_1 I)^{-1}D_k^T$ and an algorithm can be developed:

Algorithm 13 Low-rank ADI

Require: set of shift parameter $\mu_i \in \mathbb{R}^-,\ i \in \mathbb{N}$
  Define: $D_k := [C^T \quad X_k B]^T$
  $L_{k+1,1} = \sqrt{-2\mu_1}(A_k^T + \mu_1 I)^{-1} D_k^T$
  for i=2,3... **do**
    Define: $A_{k,\mu_i} = (A_k - \mu_i I)(A_k + \mu_i I)^{-1}$,
    $L_{k+1,i} = [A_{k,\mu_i}^T L_{k+1,i-1} \quad , \quad \sqrt{-2\mu_i}(A_k^T + \mu_i I)^{-1} D_k^T]$
  end for

Comparing the low-rank ADI method and the low-rank Smith method shows that both method are identical except for the used shift parameter. This relation is not surprising due to the similar structure of the ADI and Smith method. Therefore the dimension of the iterates $L_{k+1,i} \in \mathbb{R}^{n \times i(l+m)}$ is increasing, analogous to the low-rank Smith method.
Since the original algorithms and the low-rank versions always compute the same iterates $X_{k+1,i} = L_{k+1,i} L_{k+1,i}^T$, they are mathematically equivalent. Therefore the convergence results of the original algorithms are also valid for the low-rank versions and the same shift parameter selection methods are applicable.
One drawback of this low-rank ADI method can be seen in the increasing computational costs during the iteration. Remember $L_{k+1,i} \in \mathbb{R}^{n \times i(l+m)}$ and therefore the numerical effort for the computation of $L_{k+1,i}$ increases linearly. In case of a high number of iterations the advantages of the low-rank versions diminish. This problem has been solved in [51].
Li and White utilize the commutativity of the matrix pairs belonging to A_{k,μ_i} to improve the low-rank ADI method. In this version only the last few columns of $L_{k+1,i}$ are calculated in each inner iteration step. We shortly review the construction of the low-rank Cholesky factor ADI iteration (LRCF-ADI) as presented in [51],[7]:
We repeat the definition of $A_{k,\mu_i} = (A_k - \mu_i I)(A_k + \mu_i I)^{-1}$, the low-rank ADI step

$$L_{k+1,i} = [(A_k^T + \mu_i I)^{-1}(A_k^T - \mu_i I) L_{k+1,i-1} \quad , \quad \sqrt{-2\mu_i}(A_k^T + \mu_i I)^{-1} D_k^T]$$

and we denote the matrices belonging to $A_{k,\mu_v}^T, v \in \mathbb{N}_0$ with

$$S_{k,v} := (A_k^T + \mu_v I)^{-1} \ , \quad T_{k,v} := (A_k^T - \mu_v I).$$

The commutativity of the matrix pairs belonging to A_{k,μ_v}

$$S_{k,v} S_{k,w} = S_{k,w} S_{k,v}, \quad T_{k,v} T_{k,w} = T_{k,w} T_{k,v}, \quad S_{k,v} T_{k,w} = T_{k,w} S_{k,v} \qquad \forall\ v, w$$

is obvious. With this notation we are able to rewrite the columns of the i-th low-rank ADI iterate

$$\begin{aligned}
L_{k+1,i} &= [S_{k,i}T_{k,i}L_{k+1,i-1} \quad , \quad \sqrt{-2\mu_i}S_{k,i}D_k^T] \quad \Longrightarrow \\
L_{k+1,i} &= [S_{k,i}T_{k,i}(S_{k,i-1}T_{k,i-1}L_{k+1,i-2} \ , \ \sqrt{-2\mu_{i-1}}S_{k,i-1}D_k^T) \quad , \sqrt{-2\mu_i}S_{k,i}D_k^T] \\
&\qquad\qquad \Longrightarrow \quad ... \quad \Longrightarrow \\
L_{k+1,i} &= [S_{k,i}T_{k,i}...S_{k,2}T_{k,2}S_{k,1}\sqrt{-2\mu_1}D_k^T, ..., S_{k,i}T_{k,i}S_{k,i-1}\sqrt{-2\mu_{i-1}}D_k^T, \sqrt{-2\mu_i}S_{k,i}D_k^T].
\end{aligned}$$

Since all matrix pairs commute, the iterate can be written as

$$L_{k+1,i} = [P_{k,1}...P_{k,i-1}z_{k,i} \quad , \quad ... \quad , \quad P_{k,i-2}P_{k,i-1}z_{k,i} \quad , \quad P_{k,i-1}z_{k,i} \quad , \quad z_{k,i}] \tag{3.14}$$

with

$$\begin{aligned}
z_{k,i} &:= \sqrt{-2\mu_i}S_{k,i}D_k^T = \sqrt{-2\mu_i}(A_k^T + \mu_i I)^{-1}D_k^T \\
P_{k,l} &:= \left(\frac{\sqrt{-2\mu_l}}{\sqrt{-2\mu_{l+1}}}\right) S_{k,l}T_{k,l+1} \qquad \forall l \\
&= \left(\frac{\sqrt{-2\mu_l}}{\sqrt{-2\mu_{l+1}}}\right) (A_k^T + \mu_l I)^{-1}(A_k^T - \mu_{l+1}I) \\
&= \left(\frac{\sqrt{-2\mu_l}}{\sqrt{-2\mu_{l+1}}}\right) (I - (\mu_{l+1} + \mu_l))(A_k^T + \mu_l I)^{-1}.
\end{aligned}$$

The order of appearance of the shift parameter has no impact on the actual iterate $L_{k+1,i}$ therefore it is possible to reverse the index $1, ..., i$ in (3.14) resulting in

$$L_{k+1,i} = [P_{k,i}...P_{k,2}z_{k,1} \quad , \quad ... \quad , \quad P_{k,3}P_{k,2}z_{k,1} \quad , \quad P_{k,2}z_{k,1} \quad , \quad z_{k,1}]. \tag{3.15}$$

This sequence leads to following algorithm:

Algorithm 14 Low-rank Cholesky factor ADI (LRCF-ADI)

Require: set of shift parameter $\mu_i \in \mathbb{R}^-, \ i \in \mathbb{N}$
  Define: $D_k := [C^T \ \ X_k B]^T$
  $z_{k+1,1} = \sqrt{-2\mu_1}(A_k^T + \mu_1 I)^{-1}D_k^T$
  $L_{k+1,1} = z_{k+1,1}$
  for i=2,3... **do**
    Define: $P_{k,i} = \left(\frac{\sqrt{-2\mu_i}}{\sqrt{-2\mu_{i+1}}}\right)(I - (\mu_{i+1} - \mu_i))(A_k^T + \mu_i I)^{-1}$
    $z_{k+1,i} = P_{k,i}z_{k+1,i-1}$
    $L_{k+1,i} = [z_{k+1,i} \quad , \quad L_{k+1,i-1}]$
  end for

Many iterative solver for Lyapunov equations have been discussed in the previous sections. Some of them are no longer state-of-the-art but key elements for the higher developed algorithms. Up to now the low-rank Cholesky factor ADI method is the most efficient method for the solution of Lyapunov equation, in which a low-rank right side is provided. A professional implementation of the LRCF- ADI method can be found e.g. in the M.E.S.S. package [9].

3.5 Properties of ADI and Smith method

In the context of inexact Kleinman-Newton methods, we impose several requirements on the residuals $R_k,\ k \in \mathbb{N}$ of each Newton step (2.8). These additional requirements secure the monotonicity of the iterates $X_k,\ k \in \mathbb{N}$ and therefore a more global convergence property, as demonstrated in chapter 2.3. See Theorem 2.3.3 and Theorem 2.3.4 for details on the restrictions corresponding to R_k.
One key assumptions of the monotonicity preserving convergence theory is the non-negative definiteness of the residuals, i.e. $R_k \geq 0$ for all k. In this section we show, how this requirement can be addressed for different iterative Lyapunov solvers, namely Smith method, ADI method or low-rank ADI methods.
In addition, we analyze the applicability of so-called "hot-starts" in the context of inexact Riccati equation. Here one utilizes the solution X_k of the previous Newton step as an initial iterate $X_{k+1,0}$ for the next Newton step.
Recall that at Newton iteration step k the following Lyapunov equation needs to be solved:

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) + \mathcal{F}(X_k) = 0,$$

or as in (2.8) we solve for $X = X_{k+1}$

$$XA_k + A_k^T X + S_k = 0 \tag{3.16}$$

with a stable matrix A_k

$$A_k = A - BB^T X_k \quad \text{and} \quad S_k = X_k BB^T X_k + C^T C.$$

This equation is equivalent to a Stein's equation,

$$X = A_{k,\mu}^T X A_{k,\mu} + S_{k,\mu} \tag{3.17}$$

with

$$A_{k,\mu} = (A_k - \mu I)(A_k + \mu I)^{-1}, \quad S_{k,\mu} = -2\mu(A_k + \mu I)^{-T} S_k (A_k + \mu I)^{-1}, \quad \mu \in \mathbb{R}^-,$$

as shown in Lemma 3.1.1.
We shortly review Smith method and the ADI method, already introduced in

chapter 3.1 respectively chapter 3.2.
Smith method (Algorithm 5)- here we consider a simple version with one shift - is a fixed point iteration for equation (3.17) for given starting value $X_{k+1,0}$

$$X_{k+1,i} = A_{k,\mu}^T X_{k+1,i-1} A_{k,\mu} + S_{k,\mu}, \quad i = 1, 2, ... \quad \text{and } \mu < 0 \text{ fixed.}$$

ADI method (Algorithm 7) is a fixed point iteration for equation (3.17) for given starting value $X_{k+1,0}$

$$X_{k+1,i} = A_{k,\mu_i}^T X_{k+1,i-1} A_{k,\mu_i} + S_{k,\mu_i}, \quad i = 1, 2, ... \quad \text{and } \mu_i < 0 \text{ varies.} \tag{3.18}$$

Note, that the iterates to determine X_{k+1} are called $X_{k+1,i},\ i \in \mathbb{N}_0$.
In practice, cyclic versions of both methods, where a finite set of shift parameter $\mu_1, ..., \mu_s$ is used in a cyclic manner, became quite popular, see e.g. [60] and [31]. Since the Smith method and the cyclic versions are special cases of the ADI method, we consider the ADI method in the following statements.

Lemma 3.5.1. *Let Z_k be the solution of the Lyapunov equation (3.16) and let $X_{k+1,i}$ be an iterate of the ADI method. Then*

$$X_{k+1,i} - Z_k = A_{k,\mu_i}^T ... A_{k,\mu_1}^T (X_{k+1,0} - Z_k) A_{k,\mu_1} ... A_{k,\mu_i}. \tag{3.19}$$

Proof. Recall that by Lemma 3.1.1 Z_k satisfies a Stein's equation for any $\mu \in \mathbb{R}^-$ - hence for all $\mu_i \in \mathbb{R}^-$ in the ADI method

$$Z_k = A_{k,\mu_i}^T Z_k A_{k,\mu_i} + S_{k,\mu_i} \quad i = 0, 1, ...$$

Therefore we have for any iteration i with above identity

$$\begin{aligned} X_{k+1,i} - Z_k &= A_{k,\mu_i}^T X_{k+1,i-1} A_{k,\mu_i} + S_{k,\mu_i} - (A_{k,\mu_i}^T Z_k A_{k,\mu_i} + S_{k,\mu_i}) \\ &= A_{k,\mu_i}^T (X_{k+1,i-1} - Z_k) A_{k,\mu_i}. \end{aligned}$$

If we apply this identity to $X_{k+1,i-1} - Z_k$ consecutively, then we obtain the statement of the Lemma. □

The error structure in (3.19) enables us also to understand the shift parameter selection problem for ADI methods. In order to reduce the error one has to minimize the spectral radius of the matrix $A_{k,\mu_1} ... A_{k,\mu_i}$, in case of i applied parameters. This minimization problem for the shift parameter is called ADI minimax problem and an optimal set of shift parameter is given by

$$\{\mu_1, ..., \mu_s\} = \min_{\{\mu_1,...,\mu_s\} \in (-\infty, 0)} \max_{\lambda \in \sigma(A_k)} |\prod_{j=1}^{s} \frac{\lambda - \mu_j}{\lambda + \mu_j}|. \tag{3.20}$$

This problem has been intensively studied and contributions to the parameter selection problem can be found in [8, 67] and the references therein.
To estimate the residual of the Lyapunov equation using some iterate from the ADI method, we introduce the following Lemma.

Lemma 3.5.2. *Let $X_{k+1,i}$ be an iterate of the ADI method, then for the residuals of the Lyapunov equation we obtain*

$$\begin{aligned} R_k^{(i)} &:= X_{k+1,i}A_k + A_k^T X_{k+1,i} + S_k \\ &= A_{k,\mu_i}^T ... A_{k,\mu_1}^T (X_{k+1,0}A_k + A_k^T X_{k+1,0} + S_k) A_{k,\mu_1} ... A_{k,\mu_i}. \end{aligned} \tag{3.21}$$

If, in particular, the initial residual $R_k^{(0)}$ is positive (semi)definite, then all residuals $R_k^{(i)}$ are also positive (semi)definite.

Proof. Note that

$$\begin{aligned} X_{k+1,i}A_k + A_k^T X_{k+1,i} + S_k &= X_{k+1,i}A_k + A_k^T X_{k+1,i} - Z_k A_k - A_k^T Z_k \\ &= (X_{k+1,i} - Z_k)A_k + A_k^T (X_{k+1,i} - Z_k). \end{aligned}$$

Next we insert (3.19) to obtain

$$\begin{aligned} X_{k+1,i}A_k + A_k^T X_{k+1,i} + S_k = A_{k,\mu_i}^T ... A_{k,\mu_1}^T (X_{k+1,0} - Z_k) A_{k,\mu_1} ... A_{k,\mu_i} A_k \\ + A_k^T A_{k,\mu_i}^T ... A_{k,\mu_1}^T (X_{k+1,0} - Z_k) A_{k,\mu_1} ... A_{k,\mu_i}. \end{aligned}$$

Since A_k and $A_{k,\mu}$ commute for any $\mu \in \mathbb{R}^-$ and the definition of S_k, we have

$$\begin{aligned} X_{k+1,i}A_k + A_k^T X_{k+1,i} + S_k = A_{k,\mu_i}^T ... A_{k,\mu_1}^T ((X_{k+1,0} - Z_k)A_k \\ + A_k^T (X_{k+1,0} - Z_k)) A_{k,\mu_1} ... A_{k,\mu_i} \end{aligned}$$

from which (3.21) follows. From this equation we obtain the result, that if the initial residual is non-negative definite or positive definite, then this also holds for all residuals in the Lyapunov equation using any ADI iterate. □

In particular with the zero starting matrix we get:

Lemma 3.5.3. *Let $X_{k+1,0} = 0$. Then the residuals of (3.2) for the ADI iterates satisfy*

$$R_k^{(i)} \geq 0 \ \forall \ i \in \mathbb{N}.$$

Proof. The residuals of equation (3.2) for the iterates $X_{k+1,i}$ of the ADI method are given by Lemma (3.5.2)

$$R_k^{(i)} = X_{k+1,i}A_k + A_k^T X_{k+1,i} + S_k = A_{k,\mu_i}^T ... A_{k,\mu_1}^T S_k A_{k,\mu_1} ... A_{k,\mu_i} \geq 0$$

since $S_k = X_k B B^T X_k + C^T C \geq 0$. □

In summary, the key assumption of the monotonicity preserving convergence theory, that is the non-negative definiteness of the residuals, can be obtained with help of Lemma 3.5.2 and Lemma 3.5.3. As long as our initial iterate for the Lyapunov solver provides a non-negative definite residual all subsequent iterates will also lead to non-negative definite iterates. The choice of the zero matrix as initial iterate is common practice and due to Lemma 3.5.3 no further verification is necessary in this case.
The other requirements of the monotonicity preserving theory are difficult to fulfill, e.g. $R_k^{(i)} \leq C^T C$. We introduce a Lemma for this case:

Lemma 3.5.4. *Let us consider a cyclic ADI method with a finite set of shift parameter $\mu_1, ..., \mu_s \in \mathbb{R}^-$. If $C^T C$ is positive definite and $X_{k+1,0} = 0 \in \mathbb{R}^{n,n}$, there is $i_k \in \mathbb{N}$ such that for all $i \geq i_k$*

$$0 \leq R_k^{(i)} \leq C^T C$$

holds.

Proof. $R_k^{(i)} \geq 0$ is proved in the previous Lemma. Furthermore, if $C^T C > 0$, there exists $\zeta > 0$ such that for all $x \in \mathbb{C}^n$

$$\bar{x}^T C^T C x \geq \zeta \|x\|_2^2.$$

We have $\rho(A_{k,\mu}) = \max\limits_{\lambda \in \sigma(A_k)} |\frac{\lambda - \mu}{\lambda + \mu}| < 1$ for every $\mu \in \mathbb{R}^-$. Due to the special structure of the matrices $A_{k,\mu}, \mu \in \mathbb{R}^-$ it follows that

$$\rho(A_{k,\mu_1} ... A_{k,\mu_s}) = \max_{\lambda \in \sigma(A_k)} |\prod_{j=1}^{s} \frac{\lambda - \mu_j}{\lambda + \mu_j}| \leq \prod_{j=1}^{s} \max_{\lambda \in \sigma(A_k)} |\frac{\lambda - \mu_j}{\lambda + \mu_j}| < 1.$$

Therefore a consistent matrix norm $\|\cdot\|_*$ exists with $\|A_{k,\mu_1} ... A_{k,\mu_s}\|_* < 1$.
For i large enough (depending on k) we obtain with $m := i \mod (s+1)$

$$\begin{aligned}
\|R_k^{(i)}\|_2 = \|A_{k,\mu_m}^T ... A_{k,\mu_1}^T \underbrace{A_{k,\mu_s}^T ... A_{k,\mu_1}^T ... A_{k,\mu_s}^T ... A_{k,\mu_1}^T}_{\lfloor \frac{i}{s+1} \rfloor \text{ times}} S_k \\
\underbrace{A_{k,\mu_1} ... A_{k,\mu_s} ... A_{k,\mu_1} ... A_{k,\mu_s}}_{\lfloor \frac{i}{s+1} \rfloor \text{ times}} A_{k,\mu_1} ... A_{k,\mu_m} \|_2 \\
\leq c \|A_{k,\mu_1} ... A_{k,\mu_s}\|_2^{2\lfloor \frac{i}{s+1} \rfloor} \|S_k\|_2 \\
\leq c_* \|A_{k,\mu_1} ... A_{k,\mu_s}\|_*^{2\lfloor \frac{i}{s+1} \rfloor} \leq \zeta.
\end{aligned}$$

Hence for all $x \in \mathbb{C}^n$

$$\bar{x}^T R_k^{(i)} x \leq \|x\|_2^2 \|R_k^{(i)}\|_2 \leq \zeta \|x\|_2^2 \leq \bar{x}^T C^T C x$$

which is to be shown. □

According to remark (2.17) it might be possible to introduce a weaker requirement compared to the positive definiteness of the matrix C^TC to achieve the same results.
Another popular choice for the initial iterate is a so-called "hot start". Here one utilizes X_k, the solution of the previous Newton step, as initial iterate for the Lyapunov equation

$$X_{k+1}(A - BB^TX_k) + (A - BB^TX_k)^TX_{k+1} = -X_kBB^TX_k - C^TC.$$

Since the Newton iterates $X_k,\ k \in \mathbb{N}$ are convergent to some limit matrix, $X_{k+1,0} = X_k$ should be close to X_{k+1} and therefore a better initial guess compared to the zero matrix.
In the monotonicity preserving convergence theory, also the "hot start" initial iterate $X_{k+1,0} = X_k$ should provide non-negative definite residuals.

Lemma 3.5.5. *Let $X_{k+1,0} = X_k$, where X_k is the solution of the previous Newton step. Then the residuals of (3.2) for the ADI iterates satisfy*

$$R_k^{(i)} \leq (\ \geq)0 \quad \forall\ \ i \in \mathbb{N}$$

if and only if $\mathcal{F}(X_k) \leq (\ \geq)0$.

Proof. The residuals of equation (3.2) for the iterates $X_{k+1,i}$ of the ADI method are given by Lemma (3.5.2)

$$\begin{aligned} R_k^{(i)} &= X_{k+1,i}A_k + A_k^TX_{k+1,i} + S_k \\ &= A_{k,\mu_i}^T ... A_{k,\mu_1}^T (X_kA_k + A_k^TX_k + S_k)A_{k,\mu_1} ... A_{k,\mu_i}. \end{aligned} \tag{3.22}$$

Substituting the abbreviation (2.6) of A_k respectively (3.2) of S_k , we get

$$X_kA_k + A_k^TX_k + S_k = A^TX + XA - XBB^TX + C^TC = \mathcal{F}(X_k).$$

Together with (3.22), we conclude the statement of the Lemma. □

As a result, the positive (negative) semidefiniteness of the residuals is always dependent on $\mathcal{F}(X_k)$ and not can be stated beforehand. Only the concavity of the mapping $\mathcal{F}$ provides some information.

Lemma 3.5.6. *Let $\mathcal{F}$ describe the algebraic Riccati equation and assume that X_{k+1} has been determined via a step of the inexact Newton's method (2.7) with R_k. Then $\mathcal{F}(X_{k+1}) \leq R_k$.*

Proof. We outlined a statement about the concavity of $\mathcal{F}$ in Theorem 2.3.1

$$\mathcal{F}(X_{k+1}) - \mathcal{F}(X_k) \leq \mathcal{F}'(X_k)(X_{k+1} - X_k).$$

Together with the inexact Newton step (2.7) we obtain

$$\mathcal{F}(X_{k+1}) \leq R_k. \tag{3.23}$$

□

"Hot starts" usually lead to an improved performance of the Newton method but its applicability in the monotonicity preserving convergence theory can not be guaranteed in theory. In order to obtain non-negative definite residuals, as required e.g. in Theorem 2.3.4, one has to postulate the non-negative definiteness of $\mathcal{F}(X_k),\ k \in \mathbb{N}$, which is a non-trivial condition.

Chapter 4

Numerical Results

In this chapter we analyze the practical benefits of the inexact Kleinman-Newton methods, introduced in chapter 2. Our goal is to compare the exact Kleinman-Newton method (Algorithm 1) with different inexact versions, which show a linear, superlinear or even quadratic rate of local convergence, corresponding to Theorem 2.2.1.
Since all examples are connected with linear quadratic control (LQR) problems, we briefly introduce those kind of optimal control problems in section 4.1.
Of course, the performance of the exact and inexact Kleinman-Newton methods is strongly dependent on the applied Lyapunov solver. In order to show the applicability of the main iterative solvers, we present their behavior for one example, taken from [55]. Here we consider a two-dimensional optimal control problem with parabolic partial differential equation including convection. All convergence results are summarized in section 4.2.
Our second example is part of the LyaPack Users Guide [58] and describes a two dimensional heating problem. In contrast to the first example, here no convection is taken into account. Convergence properties of the inexact Kleinman-Newton methods are presented in section 4.3.
A third example is also taken from the LyaPack Users Guide [58] and is often discussed in the literature [72, 58, 66, 5, 10]. Here an algebraic Riccati equation arises in the context of an optimal control problem for the cooling process of steel profiles in a rolling mill.
Note that we concentrate on the local convergence properties and not on the monotonicity of the iterates. Further research should focus on the development of applicable numerical test for non-negative definiteness, which is required in the monotonicity preserving convergence theory (Theorem 2.3.4).

4.1 Linear quadratic regulator problems

A major field of application for algebraic Riccati equations are time-invariant linear quadratic regulator (LQR) problems. These optimal control problems play a crucial role in control theory and are therefore analyzed in detail, see [46, 2, 54, 48], to mention only a few. Since all our numerical examples arise in the context of LQR problems, we briefly introduce some relevant theory.
We consider time-invariant systems of the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \quad t > 0, \qquad x(0) = x_0 \\ y(t) &= Cx(t), \qquad\qquad\quad t > 0, \end{aligned} \tag{4.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{l\times n}$ describe the system matrices. Here $x(t) \in \mathbb{R}^n$ denotes the states, $u(t) \in \mathbb{R}^m$ the control (or input) and $y(t) \in \mathbb{R}^l$ the output of the system 4.1 for a given time t.
In addition, in order to design an optimal control problem the introduction of a performance index is necessary. Within the LQR case, one utilizes quadratic cost functionals of the form

$$J(u) = \frac{1}{2}\int_0^\infty \left(y(t)^T Q y(t) + u(t)^T R u(t)\right) dt \tag{4.2}$$

with a non-negative definite matrix $Q \in \mathbb{R}^{l\times l}$ and a positive definite matrix $R \in \mathbb{R}^{m\times m}$, which can be both interpreted as weighting matrices.
With these introductory remarks the LQR problem reads as follows.

Definition 4.1.1. *The linear quadratic regulator (LQR) problem over an infinite time horizon is defined as*

$$\begin{aligned} \min_{u\in L^2_m(0,\infty)} J(u, x_0) &= \frac{1}{2}\int_0^\infty \left(y(t)^T Q y(t) + u(t)^T R u(t)\right) dt \\ s.t. \quad \dot{x}(t) &= Ax(t) + Bu(t), \quad t > 0, \qquad x(0) = x_0 \\ y(t) &= Cx(t), \qquad\qquad t > 0, \end{aligned}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$. $Q \in \mathbb{R}^{l\times l}$ is assumed to be non-negative definite and $R \in \mathbb{R}^{m\times m}$ is a positive definite matrix.

Under suitable conditions on the system matrices, related to Assumption 2.1.2, the existence and uniqueness of an optimal control is guaranteed. Moreover the optimal control $u_*(t)$ is given by a feedback law, namely

$$u_*(t) = -R^{-1}B^T X_\infty x(t), \qquad t > 0, \tag{4.3}$$

where X_∞ is defined as the stabilizing solution of the algebraic Riccati equation

$$A^T X + XA - XBR^{-1}B^T X + C^T QC = 0. \tag{4.4}$$

In practice, the weighting matrices are often predefined as identity matrix or are provided in a factored form $Q = \tilde{Q}^T\tilde{Q}$ respectively $R = \tilde{R}\tilde{R}^T$. Therefore one can reformulate equation (4.4) into the standard formulation (1.1), which has been analyzed in the previous chapters.
In summary, for the solution of the LQR problem over an infinite time horizon an algebraic Riccati equation of type (1.1) needs to be solved. Since many optimal control problems including partial differential equations can be simplified due to linearization to LQR problems, this field has gained a lot of attention.
We utilize several examples of LQR problems, to demonstrate the numerical benefits of the inexact Kleinman-Newton methods, developed in chapter 2. These inexact methods provide an alternative to the standard Kleinman-Newton method (Algorithm 1), which is often used for the solution of LQR problems.

4.2 Two dimensional heating problem including convection

We consider an example arising from optimal control problems. The example has been considered by Morris and Navasca [55] and is described as an optimal control problem with a 2-dimensional parabolic partial differential equations including convection:

$$\min_u J(u) = \frac{1}{2}\int_0^\infty \left(\| Cz(t) \|_2^2 + \| u(t) \|_2^2\right) dt$$

s. t.

$$\begin{aligned} \frac{\partial z}{\partial t} = \frac{\partial z}{\partial x^2} + \frac{\partial z}{\partial y^2} + 20\frac{\partial z}{\partial y} + 100z = f(x,y)u(t) \qquad & (x,y) \in \Omega \\ z(x,y,t) = 0 \qquad & (x,y) \in \partial\Omega \qquad \forall t \end{aligned}$$

with $\Omega = (0,1) \times (0,1)$ and

$$f(x,y) := \begin{cases} 100 & 0.1 < x < 0.3 \quad \& \quad 0.4 < y < 0.6, \\ 0 & else. \end{cases}$$

The discretization is carried out on a 23×23 grid and central differences are used for the approximation, which results in 279 841 unknown. We choose $C =$

$(0.1, ..., 0.1)$ and $X_0 = 0$. For a detailed discussion on different values of C, see also section 4.2.4. The optimal matrix X_∞ has been computed beforehand with a higher accuracy.
In the following subsections we want to test the performance of inexact Kleinman-Newton methods utilizing different iterative Lyapunov solver. Remember that each Newton step is equivalent to the solution of a corresponding Lyapunov equation

$$X_{k+1}(A - BB^T X_k) + (A - BB^T X_k)^T X_{k+1} = -X_k BB^T X_k - C^T C. \qquad (4.5)$$

According to Theorem 2.2.1 we test three inexact Kleinman-Newton versions with an expected linear, superlinear or quadratic rate of local convergence.
Many iterative solvers for Lyapunov equations are presented in the literature, e.g. Smith method [70], ADI method [52] and low-rank ADI methods [59],[51]. Other iterative methods can be found in [31], [69], [75] or [60]. In order to give an extensive overview of the benefits of the inexact Kleinman-Newton method, we implemented the most important iterative solver, namely Smith method, the ADI method and a low-rank Cholesky factor ADI version to solve the Newton steps.
The shift parameter for the iterative methods are determined with a heuristic introduced by Penzl [60], which has been extended to a real valued version in [67]. Both shift parameter selection methods have been implemented in the M.E.S.S. package [9]. All computations were done within MATLAB.
We compare the exact Kleinman-Newton method and the inexact versions with respect to multiple goals. Our comparison includes the number of the Newton steps (outer) and the number of inner iterations (inner), which are needed to solve each Newton step. In addition, the cumulative number of all inner iterations (cumul) is also of interest. Of course, the required CPU times play a crucial role in contrasting the exact and the inexact versions.
In table 4.1 we present in advance comparative CPU times for all mentioned methods.

Lyapunov solver	Stopping criteria							
	Exact K-N		Linear		Superlinear		Superlinear/quadratic	
	$\| \mathcal{F}(X_\infty) \|$	Time	$\| \mathcal{F}(X_\infty) \|$	Time	$\| \mathcal{F}(X_\infty) \|$	Time	$\| \mathcal{F}(X_\infty) \|$	Time
Smith	3.133e-008	135.55	3.345e-008	49.91	8.946e-008	36.55	1.246e-013	46.95
ADI	3.222e-008	101.05	5.560e-008	60.80	1.860e-010	53.47	3.509e-011	50.52
LRCF-ADI	3.222e-008	7.49	5.560e-008	6.55	1.859e-010	5.54	3.509e-011	5.56

Table 4.1: Comparison of computing time

The low-rank ADI method utilizes programs of the highly developed M.E.S.S. [9] package. Detailed results on the convergence properties for the alternative iterative solver can be found in the next subsections.

4.2.1 Smith method

Here we implement Smith method (Algorithm 5) for the solution of all Lyapunov equations, occurring in the (inexact) Newton steps. Table 4.2 presents the results of the standard Kleinman-Newton method (Algorithm 1), where an accuracy of $1e-08$ for the inner iteration was predefined, i.e. the inner iteration for the k-th Newton step will be stopped if $\|R_k\| \leq 1e-08$ is satisfied. R_k, already defined e.g. in (2.8), defines the residuals of the k- inexact Newton step.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	90	90	7.639e+005	3.495e+001	1.056e+003	3.190e+004
2	212	302	1.911e+005	2.333e+001	6.677e-001	1.910e-002
3	131	433	4.794e+004	1.755e+001	7.521e-001	3.223e-002
4	91	524	1.213e+004	1.453e+001	8.279e-001	4.718e-002
5	63	587	3.172e+003	1.245e+001	8.571e-001	5.900e-002
6	59	646	8.973e+002	9.594e+000	7.704e-001	6.186e-002
7	56	702	2.357e+002	4.481e+000	4.671e-001	4.869e-002
8	50	752	1.801e+001	5.031e-001	1.123e-001	2.505e-002
9	45	797	8.544e-002	4.162e-003	8.272e-003	1.644e-002
10	45	842	8.230e-004	1.263e-005	3.036e-003	7.295e-001
11	45	887	3.133e-008	5.500e-010	4.353e-005	3.445e+000

Table 4.2: Smith method: Exact Kleinman-Newton method

An exact Kleinman-Newton method with Smith method as Lyapunov solver requires 11 Newton steps and 887 steps of Smith method to find a solution of the algebraic Riccati equation with $\| \mathcal{F}(X_\infty) \| = 3.133e-008$.
The fraction $\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$ respectively $\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$ can be utilized to estimate the local rate of convergence. In case of the exact Kleinman-Newton methods a quadratic rate of convergence is indicated.
According to Theorem 2.2.1 we develop a first inexact variant. We choose $\|R_k\| \leq 0.1 * \|\mathcal{F}(X_k)\|$ as stopping criterion for the k- Newton step and expect a linear rate of convergence. All convergence properties for this version are presented in Table 4.3, which confirm our assumption.

On the one hand, the total number of Smith iteration is 281 and therefore noticeable smaller as in the exact case. On the other hand, the linear convergent inexact version requires more Newton steps, compared with the exact method, which results in an additional numerical effort.
The total effect of these two contrary components can not be estimated beforehand. Therefore it is essential to take comparable CPU times into account. Table 4.1 presents CPU times for all discussed stopping criteria.
By selecting the stopping criterion $\|R_k\| \leq \|\mathcal{F}(X_k)\| * k^{-3}$, we obtain an inexact

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	35	35	7.637e+005	3.494e+001	1.056e+003	3.189e+004
2	15	50	1.565e+005	1.536e+001	4.395e-001	1.258e-002
3	9	59	3.202e+004	7.822e+000	5.094e-001	3.317e-002
4	7	66	6.895e+003	6.143e+000	7.854e-001	1.004e-001
5	6	72	1.554e+003	6.341e+000	1.032e+000	1.680e-001
6	9	81	4.563e+002	6.874e+000	1.084e+000	1.710e-001
7	1	82	2.839e+001	1.035e-001	1.506e-002	2.190e-003
8	10	92	2.054e+001	5.267e-001	5.089e+000	4.918e+001
9	1	93	1.404e+000	2.007e-002	3.811e-002	7.236e-002
10	11	104	9.173e-002	4.105e-004	2.045e-002	1.019e+000
11	17	121	6.622e-003	5.227e-005	1.273e-001	3.102e+002
12	22	143	5.338e-004	4.948e-006	9.467e-002	1.811e+003
13	27	170	4.642e-005	4.150e-007	8.387e-002	1.695e+004
14	32	202	4.398e-006	3.825e-008	9.216e-002	2.221e+005
15	37	239	3.694e-007	3.348e-009	8.753e-002	2.288e+006
16	42	281	3.345e-008	2.955e-010	8.827e-002	2.637e+007

Table 4.3: Smith method: Inexact K-N method with linear convergence $\eta_k = 0.1$

Kleinman-Newton method with a superlinear rate of convergence. Its convergence properties are illustrated in Table 4.4.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	28	28	7.621e+005	3.486e+001	1.053e+003	3.182e+004
2	14	42	1.505e+005	1.457e+001	4.178e-001	1.199e-002
3	13	55	3.527e+004	1.099e+001	7.547e-001	5.182e-002
4	13	68	8.756e+003	1.187e+001	1.080e+000	9.820e-002
5	12	80	2.313e+003	1.141e+001	9.616e-001	8.102e-002
6	15	95	6.714e+002	8.563e+000	7.503e-001	6.574e-002
7	16	111	1.559e+002	3.222e+000	3.762e-001	4.394e-002
8	14	125	6.525e+000	2.160e-001	6.703e-002	2.080e-002
9	17	142	3.560e-002	1.029e-003	4.763e-003	2.206e-002
10	28	170	1.332e-004	2.143e-006	2.083e-003	2.025e+000
11	40	210	8.946e-008	7.722e-010	3.603e-004	1.682e+002

Table 4.4: Smith method: Inexact K-N method with superlinear convergence $\eta_k = k^{-3}$

As in the exact case, 11 Newton steps are necessary to compute a solution of the algebraic Riccati equation. In contrast to the exact Kleinman-Newton method only 210 steps of Smith method are needed.
Finally we develop a stopping criterion resulting in an inexact Kleinman-Newton method with a quadratic rate of local convergence. We obtain the new inexact version due to a combination of two alternative stopping criteria. As long as $\|\mathcal{F}(X_k)\| \geq 1$ we use the criterion $\|R_k\| \leq \|\mathcal{F}(X_k)\| * k^{-3}$, for $\|\mathcal{F}(X_k)\| < 1$ we implement $\|R_k\| \leq \|\mathcal{F}(X_k)\|^2$ to stop the inner iteration in the k- th Newton step.

For $\|\mathcal{F}(X_k)\| >> 1$ the second stopping criterion would be useless and therefore above distinction is meaningful. The convergence results of this combination can be found in Table 4.5.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	28	28	7.621e+005	3.486e+001	1.053e+003	3.182e+004
2	14	42	1.505e+005	1.457e+001	4.178e-001	1.199e-002
3	13	55	3.527e+004	1.099e+001	7.547e-001	5.182e-002
4	13	68	8.756e+003	1.187e+001	1.080e+000	9.820e-002
5	12	80	2.313e+003	1.141e+001	9.616e-001	8.102e-002
6	15	95	6.714e+002	8.563e+000	7.503e-001	6.574e-002
7	16	111	1.559e+002	3.222e+000	3.762e-001	4.394e-002
8	14	125	6.525e+000	2.160e-001	6.703e-002	2.080e-002
9	17	142	3.560e-002	1.029e-003	4.763e-003	2.206e-002
10	21	163	7.426e-004	5.850e-006	5.686e-003	5.528e+000
11	37	200	3.694e-007	3.334e-009	5.700e-004	9.744e+001
12	69	269	1.246e-013	3.862e-016	1.158e-007	3.474e+001

Table 4.5: Smith method: Inexact K-N method with superlinear/quadratic convergence

The inexact Kleinman-Newton method shows a quadratic rate of local convergence. In addition, this version requires 12 Newton steps and 269 Smith iteration for the solution of the ARE. Here one computes a solution with $\| \mathcal{F}(X_\infty) \| = 1.246e-013$.
As a result, all inexact variants have a notable reduction of the total number of inner iterations in common. Standard Kleinman-Newton requires 887 steps of Smith methods, whereas the inexact versions need only 281, 210 respectively 269 steps. This is accompanied with a clear reduction of the numerical effort. Contrary to this result, some inexact variants need more Newton steps for the solution of the algebraic Riccati equation. Of course, the computation of additional Newton steps requires also CPU time. Therefore the total effect can not be estimated beforehand and we should take comparable CPU times (Table 4.1) into account.
Here all inexact versions show a clear reduction on the required CPU time and all inexact Kleinman-Newton method are superior to the exact version.

4.2.2 ADI method

In this section we utilize the ADI method (Algorithm 6) for the solution of the Newton steps. We implement the same stopping criteria, which have been introduced in chapter 4.2.1. Again, we compare the convergence results of the exact Kleinman-Newton method (Table 4.6) with three inexact versions, providing a linear (Table 4.7), superlinear (Table 4.8) or quadratic (Table 4.9) rate of local

convergence.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	25	25	7.639e+005	3.495e+001	1.056e+003	3.190e+004
2	22	47	1.911e+005	2.333e+001	6.677e-001	1.910e-002
3	24	71	4.794e+004	1.755e+001	7.521e-001	3.223e-002
4	25	96	1.213e+004	1.453e+001	8.279e-001	4.718e-002
5	24	120	3.172e+003	1.245e+001	8.571e-001	5.900e-002
6	24	144	8.973e+002	9.594e+000	7.704e-001	6.186e-002
7	22	166	2.357e+002	4.481e+000	4.671e-001	4.869e-002
8	28	194	1.801e+001	5.031e-001	1.123e-001	2.505e-002
9	35	229	8.544e-002	4.162e-003	8.272e-003	1.644e-002
10	41	270	8.230e-004	1.263e-005	3.036e-003	7.295e-001
11	42	312	3.222e-008	5.686e-010	4.500e-005	3.562e+000

Table 4.6: ADI method: Exact Kleinman-Newton method

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	13	13	7.639e+005	3.495e+001	1.056e+003	3.190e+004
2	2	15	1.908e+005	1.178e+001	3.371e-001	9.646e-003
3	4	19	4.767e+004	1.317e+001	1.118e+000	9.486e-002
4	3	22	1.180e+004	9.809e+000	7.449e-001	5.656e-002
5	3	25	2.934e+003	9.905e+000	1.010e+000	1.030e-001
6	4	29	8.046e+002	9.328e+000	9.418e-001	9.508e-002
7	4	33	1.962e+002	4.194e+000	4.496e-001	4.819e-002
8	1	34	3.580e+000	1.240e-002	2.956e-003	7.049e-004
9	6	40	1.183e+000	5.133e-002	4.140e+000	3.340e+002
10	7	47	4.075e-002	4.338e-004	8.453e-003	1.647e-001
11	15	62	1.551e-003	1.054e-005	2.429e-002	5.598e+001
12	22	84	1.008e-004	4.702e-007	4.462e-002	4.235e+003
13	26	110	9.144e-006	5.892e-008	1.253e-001	2.666e+005
14	32	142	8.452e-007	3.691e-009	6.264e-002	1.063e+006
15	37	179	5.560e-008	3.437e-010	9.310e-002	2.522e+007

Table 4.7: ADI method: Inexact K-N method with linear convergence $\eta_k = 0.1$

As in the Smith case, all inexact versions compute the solution of the algebraic Riccati equation with a notable smaller number of ADI iterations. The exact Kleinman-Newton method requires 312 ADI steps, the three inexact versions need only 179, 157 respectively 143 steps. Again, the number of Newton steps varies for the different stopping criteria. Especially the linearly convergent version needs more Newton steps compared to all other methods. The computation of the additional Newton steps involves an additional numerical effort.
The total outcome of this two contrary effects can not be stated beforehand. CPU times are presented in Table 4.1 and indicate the benefits of the inexact Kleinman-Newton methods.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	9	9	7.639e+005	3.492e+001	1.055e+003	3.188e+004
2	2	11	1.908e+005	1.178e+001	3.374e-001	9.662e-003
3	4	15	4.767e+004	1.317e+001	1.118e+000	9.486e-002
4	3	18	1.180e+004	9.809e+000	7.449e-001	5.656e-002
5	4	22	3.119e+003	1.218e+001	1.242e+000	1.266e-001
6	6	28	8.920e+002	9.564e+000	7.850e-001	6.443e-002
7	6	34	2.334e+002	4.441e+000	4.644e-001	4.856e-002
8	7	41	1.742e+001	4.921e-001	1.108e-001	2.495e-002
9	9	50	8.525e-002	3.902e-003	7.929e-003	1.611e-002
10	23	73	8.673e-004	1.206e-005	3.090e-003	7.919e-001
11	34	107	3.936e-007	1.616e-009	1.340e-004	1.111e+001
12	50	157	1.860e-010	8.110e-013	5.020e-004	3.107e+005

Table 4.8: ADI method: Inexact K-N method with superlinear convergence $\eta_k = k^{-3}$

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	9	9	7.639e+005	3.492e+001	1.055e+003	3.188e+004
2	2	11	1.908e+005	1.178e+001	3.374e-001	9.662e-003
3	4	15	4.767e+004	1.317e+001	1.118e+000	9.486e-002
4	3	18	1.180e+004	9.809e+000	7.449e-001	5.656e-002
5	4	22	3.119e+003	1.218e+001	1.242e+000	1.266e-001
6	6	28	8.920e+002	9.564e+000	7.850e-001	6.443e-002
7	6	34	2.334e+002	4.441e+000	4.644e-001	4.856e-002
8	7	41	1.742e+001	4.921e-001	1.108e-001	2.495e-002
9	9	50	8.525e-002	3.902e-003	7.929e-003	1.611e-002
10	13	63	5.636e-003	2.071e-005	5.306e-003	1.360e+000
11	26	89	8.033e-006	5.394e-008	2.605e-003	1.258e+002
12	54	143	3.509e-011	1.980e-013	3.671e-006	6.805e+001

Table 4.9: ADI method: Inexact K-N method with superlinear/quadratic convergence

Again, all inexact version exhibit some benefits compared to the exact version, considered with respect to the required CPU time.

4.2.3 Low-rank Cholesky factor ADI method

Finally we implement one "state-of-the-art" solver, namely the low-rank Cholesky factor ADI method (Algorithm 14) for the solution of the Newton steps. In case of LRCF-ADI methods we are able to use the *lp_lradi.m* routine of the M.E.S.S. package [9] for the solution of the Lyapunov equations.
Note, the *lp_lradi.m* routine utilizes stopping criterion based on relative residuals, i.e. $\|R_k\|/\|Y_k\|$, where Y_k describes the right side of the Lyapunov equation (4.5) in the k- th Newton step. In our context we have to compute the residual R_k, e.g. defined in (2.8), and therefore a slightly modification of the *lp_lradi.m* was

necessary.
We utilize all stopping criteria, which have been introduced in section 4.2.1. Therefore we compare three inexact versions with the exact Kleinman-Newton method (Table 4.10). Corresponding to Theorem 2.2.1 we consider inexact Kleinman-Newton methods with a linear (Table 4.11), superlinear (Table 4.12) or quadratic (Table 4.13) rate of local convergence.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	25	25	7.639e+005	3.495e+001	1.056e+003	3.190e+004
2	22	47	1.911e+005	2.333e+001	6.677e-001	1.910e-002
3	24	71	4.794e+004	1.755e+001	7.521e-001	3.223e-002
4	25	96	1.213e+004	1.453e+001	8.279e-001	4.718e-002
5	24	120	3.172e+003	1.245e+001	8.571e-001	5.900e-002
6	24	144	8.973e+002	9.594e+000	7.704e-001	6.186e-002
7	22	166	2.357e+002	4.481e+000	4.671e-001	4.869e-002
8	28	194	1.801e+001	5.031e-001	1.123e-001	2.505e-002
9	35	229	8.544e-002	4.162e-003	8.272e-003	1.644e-002
10	41	270	8.230e-004	1.263e-005	3.036e-003	7.295e-001
11	42	312	3.222e-008	5.686e-010	4.500e-005	3.562e+000

Table 4.10: LRCF-ADI: Exact Kleinman-Newton method

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	13	13	7.639e+005	3.495e+001	1.056e+003	3.190e+004
2	2	15	1.908e+005	1.178e+001	3.371e-001	9.646e-003
3	4	19	4.767e+004	1.317e+001	1.118e+000	9.486e-002
4	3	22	1.180e+004	9.809e+000	7.449e-001	5.656e-002
5	3	25	2.934e+003	9.905e+000	1.010e+000	1.030e-001
6	4	29	8.046e+002	9.328e+000	9.418e-001	9.508e-002
7	4	33	1.962e+002	4.194e+000	4.496e-001	4.819e-002
8	1	34	3.580e+000	1.240e-002	2.956e-003	7.049e-004
9	6	40	1.183e+000	5.133e-002	4.140e+000	3.340e+002
10	7	47	4.075e-002	4.338e-004	8.453e-003	1.647e-001
11	15	62	1.551e-003	1.054e-005	2.429e-002	5.598e+001
12	22	84	1.008e-004	4.702e-007	4.462e-002	4.235e+003
13	26	110	9.144e-006	5.892e-008	1.253e-001	2.666e+005
14	32	142	8.452e-007	3.691e-009	6.264e-002	1.063e+006
15	37	179	5.560e-008	3.437e-010	9.310e-002	2.522e+007

Table 4.11: LRCF-ADI: Inexact K-N method with linear convergence $\eta_k = 0.1$

Since the low-rank Cholesky factor ADI method produces the same iterates as the ADI method, both show similar convergence properties. Therefore all conclusions of the previous chapter 4.2.2 are also valid in the low-rank case.
In addition, the low-rank formulation and the efficient implementation in the

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	9	9	7.639e+005	3.492e+001	1.055e+003	3.188e+004
2	2	11	1.908e+005	1.178e+001	3.374e-001	9.662e-003
3	4	15	4.767e+004	1.317e+001	1.118e+000	9.486e-002
4	3	18	1.180e+004	9.809e+000	7.449e-001	5.656e-002
5	4	22	3.119e+003	1.218e+001	1.242e+000	1.266e-001
6	6	28	8.920e+002	9.564e+000	7.850e-001	6.443e-002
7	6	34	2.334e+002	4.441e+000	4.644e-001	4.856e-002
8	7	41	1.742e+001	4.921e-001	1.108e-001	2.495e-002
9	9	50	8.525e-002	3.902e-003	7.929e-003	1.611e-002
10	23	73	8.673e-004	1.206e-005	3.090e-003	7.919e-001
11	34	107	3.936e-007	1.616e-009	1.340e-004	1.111e+001
12	50	157	1.859e-010	8.111e-013	5.020e-004	3.107e+005

Table 4.12: LRCF-ADI: Inexact K-N method with superlinear convergence $\eta_k = k^{-3}$

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	9	9	7.639e+005	3.492e+001	1.055e+003	3.188e+004
2	2	11	1.908e+005	1.178e+001	3.374e-001	9.662e-003
3	4	15	4.767e+004	1.317e+001	1.118e+000	9.486e-002
4	3	18	1.180e+004	9.809e+000	7.449e-001	5.656e-002
5	4	22	3.119e+003	1.218e+001	1.242e+000	1.266e-001
6	6	28	8.920e+002	9.564e+000	7.850e-001	6.443e-002
7	6	34	2.334e+002	4.441e+000	4.644e-001	4.856e-002
8	7	41	1.742e+001	4.921e-001	1.108e-001	2.495e-002
9	9	50	8.525e-002	3.902e-003	7.929e-003	1.611e-002
10	13	63	5.636e-003	2.071e-005	5.306e-003	1.360e+000
11	26	89	8.033e-006	5.394e-008	2.605e-003	1.258e+002
12	54	143	3.509e-011	1.980e-013	3.671e-006	6.805e+001

Table 4.13: LRCF-ADI: Inexact K-N method with superlinear/quadratic convergence

M.E.S.S. package [9] lead to a clear reduction of the CPU times (Table 4.1), compared to the standard ADI method.

4.2.4 Observation

An interesting characteristic of the inexact Kleinman-Newton methods can be observed in this example, taken from the Morris and Navasca paper [55].
Depending on the structure of C the first Newton step with $X_0 = 0$ leads away from the solution of the algebraic Riccati equation. By varying the size of C, we are able to influence the quality $\|\mathcal{F}(X_1)\|$ of the first Newton step.
We define $C = c * (1, ..., 1)$ for different $c \in \mathbb{R}$ and consider the resulting values of $\|\mathcal{F}(X_0)\|$ and $\|\mathcal{F}(X_1)\|$ in Table 4.14.
For $C = (1, ..., 1)$ the exact Kleinman-Newton algorithm produces a first iterate

c	$\|\mathcal{F}(X_0)\|$	$\|\mathcal{F}(X_1)\|$
0.01	5.290e-002	7.639e+001
0.05	1.322e+000	4.775e+004
0.1	5.290e+000	7.639e+005
0.5	1.323e+002	4.775e+008
1	5.290e+002	7.639e+009

Table 4.14

X_1 with $\|\mathcal{F}(X_1)\| = 7.639e+009$. But in this situation the exact method (Table 4.15) fails to compute a second iterate X_2 satisfying the exact inner stopping criterion, here $\|R_k\| \leq 1e-08$.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$
0	0	0	5.290e+002
1	28	28	7.639e+009
2	***	***	***

Table 4.15: LRCF-ADI: Exact Kleinman-Newton method

But it is important to notice, that this problem does not occur in case of inexact Newton's methods. We present the convergence properties of the linear (Table 4.16) and superlinear (Table 4.17) convergent inexact Kleinman-Newton method for the example with $C = (1, ..., 1)$. As Lyapunov solver we utilize the LCRF-ADI methods, the inexact stopping criteria are similar to those in section 4.2.3.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$
1	13	13	7.639e+009
2	2	15	1.910e+009
3	2	17	4.775e+008
4	2	19	1.194e+008
5	2	21	2.984e+007
6	3	24	7.452e+006
...	...	...	...
17	13	78	1.261e-002
18	17	95	1.211e-003
19	24	119	1.162e-004
20	29	148	1.099e-005
21	35	183	9.785e-007
22	42	225	4.891e-008

Table 4.16: LRCF-ADI: Inexact K-N method with linear convergence $\eta_k = 0.1$

An alternative possibility to avoid this problem would be the implementation

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$
1	9	9	7.640e+009
2	2	11	1.910e+009
3	2	13	4.775e+008
4	2	15	1.194e+008
5	2	17	2.984e+007
6	3	20	7.453e+006
7	6	26	1.888e+006
8	7	33	4.863e+005
9	7	40	1.336e+005
10	9	49	3.829e+004
11	8	57	5.419e+003
12	7	64	6.536e+001
13	14	78	3.389e+000
14	17	95	2.175e-002
15	31	126	4.483e-006
16	51	177	1.030e-009

Table 4.17: LRCF-ADI: Inexact K-N method with superlinear convergence $\eta_k = k^{-3}$

of line search methods, as introduced and analyzed in [6].
Another option would be the use of different stopping criteria depending e.g. on stagnation detection techniques or relative change based criteria. An extensive overview on these methods can be found in [67] and the references therein. Both stopping criteria are implemented in the M.E.S.S. package [9].

4.3 Two dimensional heating problem

Our second example is taken from the LyaPack User Guide [58]. We solve the algebraic Riccati equation, where the matrices A, B and C are determined with help of the *demo_r1.m* file. In contrast to the Morris and Navasca paper [55], here a two dimensional heat equation without convection is considered

$$\frac{\partial z}{\partial t} = \frac{\partial z}{\partial x^2} + \frac{\partial z}{\partial y^2} + f(x,y)u(t) \qquad (x,y) \in \Omega = (0,1) \times (0,1).$$

For details on this example see [58].
The discretization is carried out on a 30×30 grid, resulting in 810000 unknown. We select $X_0 = 0$ as initial iterate and use the LRCF-ADI method (Algorithm 14) for the solution of each Newton step. The low-rank Cholesky-factor ADI method (Algorithm 14) has been efficiently implemented in the M.E.S.S. (Matrix Equation Sparse Solver) package [9], the successor of the LyaPack Matlab Toolbox [58].

The stopping criteria are chosen such that three inexact Kleinman-Newton methods, with a expected linear, superlinear respectively quadratic rate of convergence, can be analyzed.

Table 4.18 presents the results of the exact Kleinman-Newton method, where

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	22	22	2.524e+006	9.892e+002	1.181e+001	1.411e-001
2	24	46	6.186e+005	5.303e+002	5.361e-001	5.420e-004
3	22	68	1.427e+005	2.717e+002	5.122e-001	9.659e-004
4	24	92	2.584e+004	1.011e+002	3.720e-001	1.369e-003
5	24	116	1.952e+003	1.280e+001	1.267e-001	1.254e-003
6	21	137	1.332e+001	1.206e-001	9.418e-003	7.357e-004
7	21	158	5.144e-004	5.690e-006	4.719e-005	3.914e-004
8	21	179	1.241e-009	3.461e-012	6.083e-007	1.069e-001

Table 4.18: LRCF-ADI: Exact Kleinman-Newton method

an accuracy of $1e-08$ was set for the inner iteration. A total number of 179 LRCF-ADI iteration and 8 Newton steps were necessary to compute a solution.

The linear convergent inexact method (Table 4.19) requires 12 Newton steps

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	2	2	2.496e+006	9.881e+002	1.180e+001	1.409e-001
2	4	6	6.082e+005	5.485e+002	5.550e-001	5.617e-004
3	4	10	1.391e+005	2.829e+002	5.157e-001	9.403e-004
4	4	14	2.359e+004	1.032e+002	3.649e-001	1.290e-003
5	4	18	1.371e+003	1.169e+001	1.133e-001	1.098e-003
6	6	24	5.957e+000	7.925e-002	6.778e-003	5.796e-004
7	8	32	4.574e-001	6.703e-004	8.458e-003	1.067e-001
8	10	42	2.255e-002	3.375e-005	5.035e-002	7.512e+001
9	12	54	1.075e-003	4.077e-007	1.208e-002	3.579e+002
10	15	69	4.861e-006	8.433e-009	2.068e-002	5.072e+004
11	18	87	2.401e-007	4.351e-010	5.160e-002	6.118e+006
12	20	107	1.308e-008	1.113e-011	2.559e-002	5.881e+007

Table 4.19: LRCF-ADI: Inexact K-N method with linear convergence $\eta_k = 0.1$

and only 107 steps of the LRCF-ADI methods.

As in the exact case, the inexact Kleinman-Newton method, with the superlinear rate of convergence (Table 4.20), need 8 Newton steps. In contrast to the exact methods the inexact only requires only 70 inner iterations, which is a clear reduction.

In order to obtain a local quadratic rate of convergence for an inexact Kleinman-Newton method, we have to combine two stopping criteria. For the first Newton steps we utilize the stopping criterion of the inexact method with a superlinear rate of convergence. According to Theorem 2.2.1 we stop the subsequent Newton

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	1	1	3.321e+005	1.452e+002	1.734e+000	2.071e-002
2	4	5	7.016e+004	6.055e+001	4.171e-001	2.873e-003
3	4	9	8.978e+003	1.344e+001	2.219e-001	3.664e-003
4	6	15	2.984e+002	7.896e-001	5.877e-002	4.374e-003
5	6	21	2.258e+000	3.824e-003	4.843e-003	6.134e-003
6	12	33	1.075e-003	4.062e-007	1.062e-004	2.777e-002
7	16	49	7.601e-007	2.020e-009	4.972e-003	1.224e+004
8	21	70	1.220e-009	3.441e-012	1.704e-003	8.435e+005

Table 4.20: LRCF-ADI: Inexact K-N method with superlinear convergence $\eta_k = k^{-3}$

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	1	1	3.321e+005	1.452e+002	1.734e+000	2.071e-002
2	4	5	7.016e+004	6.055e+001	4.171e-001	2.873e-003
3	4	9	8.978e+003	1.344e+001	2.219e-001	3.664e-003
4	6	15	2.984e+002	7.896e-001	5.877e-002	4.374e-003
5	6	21	2.258e+000	3.824e-003	4.843e-003	6.134e-003
6	12	33	1.075e-003	4.062e-007	1.062e-004	2.777e-002
7	22	55	6.450e-010	6.516e-013	1.604e-006	3.949e+000

Table 4.21: LRCF-ADI: Inexact K-N method with superlinear/quadratic convergence

steps ($k > 3$) with a accuracy of $0.001 * \|\mathcal{F}(X_k)\|^2$ in the k- th Newton step. Convergence results for this combination can be found in Table 4.21.
In summary, all inexact variants show a clear reduction in the total number of inner iterations. Nevertheless the CPU times, as presented in Table 4.22, indicate that not every inexact version is superior to the exact implementation. In the particular case of the inexact version with the linear rate of convergence, the additional effort to compute more Newton steps and evaluate the stopping criterion annihilates all advantages of the reduced number of inner iterations. The other inexact methods are still very effective and show advantages compared with the exact implementation.

Lyapunov solver	Stopping criteria							
	Exact K-N		Linear		Superlinear		Superlinear/quadratic	
	$\| \mathcal{F}(X_\infty) \|$	Time	$\| \mathcal{F}(X_\infty) \|$	Time	$\| \mathcal{F}(X_\infty) \|$	Time	$\| \mathcal{F}(X_\infty) \|$	Time
LRCF-ADI	1.241e-009	16.83	1.308e-008	17.85	1.220e-009	11.46	6.450e-010	10.33

Table 4.22: Comparison of computing time

4.4 Optimal cooling of steel profiles

Here we consider a practical example, which has been often discussed in the literature [72, 58, 66, 5, 10]. The goal is to optimize the cooling process of steel profiles in a rolling mill. On the one hand one wants to reduce to the temperature of the steel profiles as fast as possible. On the other hand it is necessary to take quality standards of the steel into account. A detailed description of the model equations and boundary conditions can be found e.g. in [66].
We obtain the system matrices $A \in \mathbb{R}^{821\times821}$, $B \in \mathbb{R}^{821\times6}$ and $C \in \mathbb{R}^{6\times821}$ with help of *rail821.mat*, provided by the LyaPack Users Guide [58]. Note, *rail821.mat* generates the data of a generalized dynamical system, see e.g. [67] for details on generalized systems. Penzl [58] indicates a simple technique to transfer the generalized system into a standard formulation (4.1) with help of matrix factorizations. Due to this reformulation, an algebraic Riccati equation of type (1.1) needs to be solved. We choose the weighting factor $R = \tilde{R}\tilde{R}^T$ with $\tilde{R} = 0.01I$ for our computations.
Again, we compare the convergence of the exact Kleinman-Newton method (Algorithm 1) with different inexact versions. According to Theorem 2.2.1 we choose stopping criteria resulting in a linear, superlinear and quadratic rate of convergence. We utilize the zero matrix as an initial iterate $X_0 = 0 \in \mathbb{R}^{821\times821}$ and the low-rank Cholesky factor ADI method (Algorithm 14) as iterative solver for the Lyapunov equations under consideration. The optimal solution X_∞ has been computed beforehand with a slightly higher accuracy.
Table 4.23 presents the convergence properties of the exact Kleinman-Newton method with an accuracy of $\|R_k\| \leq 1e-08$ for all Newton steps.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k-X_\infty\|}{\|X_{k-1}-X_\infty\|}$	$\frac{\|X_k-X_\infty\|}{\|X_{k-1}-X_\infty\|^2}$
1	50	50	9.446e+004	1.890e+000	2.673e+000	3.780e+000
2	30	80	2.360e+004	9.121e-001	4.826e-001	2.554e-001
3	29	109	5.888e+003	4.283e-001	4.695e-001	5.148e-001
4	34	143	1.460e+003	1.916e-001	4.474e-001	1.045e+000
5	30	173	3.537e+002	7.875e-002	4.109e-001	2.145e+000
6	33	206	7.903e+001	2.840e-002	3.607e-001	4.580e+000
7	31	237	1.368e+001	8.292e-003	2.920e-001	1.028e+001
8	31	268	1.164e+000	1.192e-003	1.437e-001	1.733e+001
9	32	300	1.640e-002	2.394e-005	2.009e-002	1.686e+001
10	32	332	4.691e-006	9.479e-009	3.960e-004	1.654e+001
11	32	364	9.445e-009	8.955e-013	9.447e-005	9.966e+003

Table 4.23: LRCF-ADI: Exact Kleinman-Newton method

The exact version fails to compute the first Newton step with the required accuracy. We therefore stopped the iteration after 50 steps of the LRCF-ADI

method. A total amount of 364 steps of the iterative solver and 11 Newton steps are necessary to compute a solution of the algebraic Riccati equation.
The inexact Newton's method with a linear rate of convergence (Table 4.24) utilizes 17 Newtons steps with 251 inner iteration for the solution of the equation.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	3	3	8.897e+004	1.820e+000	2.574e+000	3.641e+000
2	2	5	1.576e+004	6.948e-001	3.817e-001	2.097e-001
3	2	7	5.916e+003	2.948e-001	4.243e-001	6.106e-001
4	5	12	6.604e+002	1.397e-001	4.740e-001	1.608e+000
5	6	18	2.566e+002	4.511e-002	3.229e-001	2.310e+000
...	...	...	...	...	...	...
13	24	140	1.111e-005	6.403e-010	3.336e-001	1.738e+008
14	25	165	1.594e-006	2.576e-010	4.024e-001	6.284e+008
15	27	192	6.943e-007	1.754e-011	6.807e-002	2.642e+008
16	29	221	1.096e-007	4.344e-012	2.477e-001	1.413e+010
17	30	251	2.040e-008	3.110e-012	7.158e-001	1.648e+011

Table 4.24: LRCF-ADI: Inexact K-N method with linear convergence $\eta_k = 0.5$

Here we choose $\|R_k\| \leq \|\mathcal{F}(X_k)\| * 0.5$ as stopping criterion. This version needs more Newton steps but fewer iteration of the LCRF ADI method compared to the exact Kleinman-Newton method. Therefore it is not possible to evaluate the numerical benefit of the inexact version beforehand and we have to take comparable CPU times (Table 4.27) into account.
Our second inexact method stops the inner iteration at an accuracy of $\|R_k\| \leq \|\mathcal{F}(X_k)\| * k^{-2.2}$ for the k-th Newton step. According to Theorem 2.2.1 we expect a superlinear rate of convergence. Details on the convergence of this version can be found in Table 4.25.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	1	1	2.835e+004	3.700e-001	5.233e-001	7.401e-001
2	2	3	6.137e+003	2.328e-001	6.293e-001	1.701e+000
3	7	10	2.299e+003	3.440e-001	1.478e+000	6.347e+000
4	7	17	5.578e+002	1.402e-001	4.075e-001	1.185e+000
5	8	25	1.257e+002	4.952e-002	3.533e-001	2.520e+000
6	9	34	2.383e+001	1.485e-002	2.998e-001	6.054e+000
7	12	46	2.675e+000	2.896e-003	1.951e-001	1.314e+001
8	15	61	7.309e-002	1.158e-004	4.000e-002	1.381e+001
9	21	82	1.096e-004	9.535e-008	8.231e-004	7.105e+000
10	28	110	6.437e-007	8.083e-012	8.477e-005	8.890e+002
11	34	144	2.162e-009	3.541e-014	4.381e-003	5.420e+008

Table 4.25: LRCF-ADI: Inexact K-N method with superlinear convergence $\eta_k = k^{-2.2}$

11 Newton steps are required to solve the ARE but only 144 inner iteration are

computed, which is a clear reduction.
Finally we present an inexact version, which is a combination of two stopping criteria. For the first five Newton steps we utilize the same criterion as in the superlinear case. We switch to a different accuracy of $\|R_k\| \leq 0.001 * \|\mathcal{F}(X_k)\|^2$ for all subsequent Newton steps. The convergence properties of this method is outlined in Table 4.26 and a local quadratic rate of convergence can be achieved.

outer	inner	cumul	$\| \mathcal{F}(X_k) \|$	$\| X_k - X_\infty \|$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|}$	$\frac{\|X_k - X_\infty\|}{\|X_{k-1} - X_\infty\|^2}$
1	1	1	2.835e+004	3.700e-001	5.233e-001	7.401e-001
2	2	3	6.137e+003	2.328e-001	6.293e-001	1.701e+000
3	7	10	2.299e+003	3.440e-001	1.478e+000	6.347e+000
4	7	17	5.578e+002	1.402e-001	4.075e-001	1.185e+000
5	8	25	1.257e+002	4.952e-002	3.533e-001	2.520e+000
6	8	33	2.279e+001	1.416e-002	2.859e-001	5.773e+000
7	12	45	2.465e+000	2.757e-003	1.947e-001	1.375e+001
8	18	63	6.180e-002	1.017e-004	3.690e-002	1.338e+001
9	25	88	3.759e-005	7.029e-008	6.909e-004	6.791e+000
10	50	138	9.402e-010	2.880e-014	4.097e-007	5.830e+000

Table 4.26: LRCF-ADI: Inexact K-N method with superlinear/quadratic convergence

In summary, like in all other discussed examples a notable reduction in total number of inner iterations is obtained for all inexact versions. Again, the total numerical benefit can not be estimated beforehand because the number of needed Newton steps varies for the different stopping criteria. Note, the inexact version with a superlinear rate of convergence is easily comparable to the exact Kleinman-Newton method and shows a superior behaviour, which is also confirmed by the CPU times in Table 4.27.

Lyapunov solver	Stopping criteria							
	Exact K-N		Linear		Superlinear		Superlinear/quadratic	
	$\| \mathcal{F}(X_\infty) \|$	Time	$\| \mathcal{F}(X_\infty) \|$	Time	$\| \mathcal{F}(X_\infty) \|$	Time	$\| \mathcal{F}(X_\infty) \|$	Time
LRCF-ADI	9.445e-009	412.08	1.068e-008	229.36	2.162e-009	183.87	9.402e-010	202.03

Table 4.27: Comparison of computing time

Chapter 5

Feedback gain algorithms

For linear quadratic regulator (LQR) problems, namely

$$\begin{aligned} \min_u J(u) &= \frac{1}{2}\int_0^\infty \left(\| Cx(t) \|_2^2 + \| u(t) \|_2^2 \right) dt \\ s.t. \quad \dot{x}(t) &= Ax(t) + Bu(t) \qquad\qquad x(0) = x_0, \end{aligned} \tag{5.1}$$

the optimal control is known and defined as $u_*(t) = (A - BB^T X_\infty)x(t)$, where X_∞ describes a stabilizing solution of the algebraic Riccati equation

$$\mathcal{F}(X) = A^T X + XA - XBB^T X + C^T C = 0. \tag{5.2}$$

LQR problems are a major area of application for Riccati equation and are well analyzed, see e.g. [48, 46, 2].
In the special case of LQR problems one is not mainly interested in the solution X_∞ of the Riccati equation (5.2), the knowledge of the so-called feedback gain matrix $B^T X_\infty$ is sufficient to compute the optimal control u_*.
The dimension of $B^T X_\infty$ is usually much smaller than the dimension of X_∞. Feedback gain algorithms take advantage of this fact and compute the matrix $B^T X_\infty$ directly without information about X_∞. These algorithms exhibit several practical benefits compared to the standard Kleinman-Newton (Algorithm 1).
The first feedback gain algorithm has been introduced by Banks and Ito [3]. Also in the feedback gain context, Newton's method is utilized for the solution of (5.2) but here $B^T X_k$ is computed, where X_k, $k \in \mathbb{N}$ are the Newton iterates. This method is based on the factored version of the Smith method (Algorithm 11), which has been stated in chapter 3.3. We outline the development of this feedback gain algorithm in section 5.1.
In order to apply inexact Newton's methods in the feedback framework, several difficulties are encountered. We summarize them in section 5.2 and present possible solutions.

5.1 Derivation

The goal of a feedback gain algorithm is the computation of $B^T X_\infty$, in which X_∞ describes the stabilizing solution of (5.2). An obvious but unprofitable way to achieve the feedback gain matrix would be the simple computation of

$$K_k = B^T X_k \qquad k \in \mathbb{N},$$

where X_k describes the k-th Newton iterate of the Kleinman-Newton method (Algorithm 1), i.e. the solution X of

$$X(A - BB^T X_{k-1}) + (A - BB^T X_{k-1})^T X = -X_{k-1} B B^T X_{k-1} - C^T C.$$

This slight modification of the Kleinman-Newton method enables the calculation of the feedback gain matrix $B^T X_\infty = \lim_{k\to\infty} K_k$. Nevertheless, the same Lyapunov equation as in the Kleinman-Newton method have to be solved and no numerical benefit can be obtained within this modification. The calculation of the $n \times n$ Newton iterates $X_k,\ k \in \mathbb{N}$ is still necessary.
Banks and Ito [3] implemented the first meaningful feedback gain algorithm. Instead of the solution X_∞ of equation (5.2) their method allows the computation of the feedback gain matrix $B^T X_\infty$. This version is still based on Newton's method even though the original Newton iterates $X_k,\ k \in \mathbb{N}$ will no longer be taken into account.
In the original paper [3], the authors utilized a modified incremental version of Newton's method. We analyzed this formulation in section 2.4 and demonstrated its inapplicability in case of inexact Newton's methods. Nevertheless the derivation of a feedback gain algorithm can be easily transferred from the modified version to the standard Kleinman-Newton method (Algorithm 1).
The development of the feedback gain algorithm is closely related to the factored version of Smith method, presented in section 3.3. Remember the main iteration loop of Algorithm 11:

$$X_{k+1,i+1} = X_{k+1,i} - 2\mu M_{k+1,i+1}^T M_{k+1,i+1} \tag{5.3}$$

Only one further step is needed to deduce a new algorithm.
This iteration sequence can be used to implement a feedback gain algorithm, because multiplying (5.3) with B^T leads to

$$\begin{aligned} J_{k+1,i+1} =& J_{k+1,i} - 2\mu B^T M_{k+1,i+1}^T M_{k+1,i+1} \\ with \quad J_{k+1,i} :=& B^T X_{k+1,i} \qquad\qquad \forall\ i \in \mathbb{N}. \end{aligned} \tag{5.4}$$

For a definition of the appearing matrices see section 3.3 or Algorithm 11.
Above iteration loop can be used as iterative solver for each Newton step in

the Kleinman-Newton method (Algorithm 1). But here we calculate the matrix $B^T X_k,\ k \in \mathbb{N}$ instead of the actual Newton iterates $X_k,\ k \in \mathbb{N}$. In addition, the Newton iterates $X_k,\ k \in \mathbb{N}$ are not required for above computations and their calculation and storage can be omitted. Only information about $B^T X_k,\ k \in \mathbb{N}$ are utilized.
We outline the feedback version of Algorithm 11 for a better understanding.

Algorithm 15 Feedback gain oriented Lyapunov solver

Require: stable matrix A_k, D_k according to (3.8) and shift parameter $\mu \in \mathbb{R}^-$
  Define: $A_{k,\mu} = (A_k - \mu I)(A_k + \mu I)^{-1}$, $M_{k+1,1} = D_k(A_k + \mu I)^{-1}$
Ensure: $J_{k+1,0} = 0$, $J_{k+1,1} = J_{k+1,0} - 2\mu B^T M_{k+1,1}^T M_{k+1,1}$
  for i=1,2,... **do**
    $M_{k+1,i+1} = M_{k+1,i} A_{k,\mu}$
    $J_{k+1,i+1} = J_{k+1,i} - 2\mu B^T M_{k+1,i+1}^T M_{k+1,i+1}$
  end for

Algorithm 15 ensures $\lim_{i\to\infty} J_{k+1,i} = B^T X_{k+1}$. Note, the Newton iterates $X_k,\ k \in \mathbb{N}$ do never occur in the computations above, only the matrices $B^T X_k,\ k \in \mathbb{N}$ are required and calculated. Since the Newton iterates X_k converge to a solution X_∞ of the algebraic Riccati equation (5.2), we obtain $\lim_{k\to\infty} B^T X_k = B^T X_\infty$.
The factored Smith version is only a reformulation of the original Smith method. An obvious connection of the Smith iterates and the feedback gain iterates can be stated.

Lemma 5.1.1. *There is a relation between the iterates $X_{k+1,i} \in \mathbb{R}^{n\times n}$ of Smith method (Algorithm 5) and the iterates $J_{k+1,i} \in \mathbb{R}^{m\times n}$ of Algorithm 15, namely*

$$B^T X_{k+1,i} = J_{k+1,i} \quad \forall i \in \mathbb{N}. \tag{5.5}$$

Proof. The proof follows easily from the derivation of the modified Smith method, specified in section 3.3. Comparing (3.13) and (5.4) provides the desired result. □

One advantage of the feedback gain algorithms can be seen in the dimension of the iterates. We no longer work on the $n \times n$ matrices of the standard methods, only computations with the $m \times n$ matrices with $m << n$ are necessary, which results in a clear reduction of the numerical effort. Additional numerical benefits can be found in the decreased storage requirements because the storage of the $n \times n$ iterates $X_k,\ k \in \mathbb{N}$ are omitted.
Of course, Smith Method and its factored form are no longer state-of-the-art. Penzl already implemented an implicit low-rank Cholesky factor Newton method,

a highly developed algorithm for computation of the feedback gain matrix in his LyaPack Users guide [58]. Here the occurring Newton steps are solved with the low-rank Cholesky factor ADI method (Algorithm 14), which has been modified for the computation of the feedback gain matrices, see [58] or [67] for details.

5.2 Challenges of an inexact version

In order to introduce feedback gain algorithms utilizing an inexact Newton's method, several difficulties are encountered. It is impossible to apply the stopping criteria, provided by the theory about inexact Newton's methods (e.g. Theorem 2.2.1), directly.
All stopping criteria, resulting in a linear, superlinear or quadratic rate of convergence, require the computation of $\mathcal{F}(X_k)$ and the residuals R_k of the k-th Newton step. Both quantities are dependent on X_k and therefore not known in case of feedback gain algorithm because the calculation of the Newton iterates $X_k,\ k \in \mathbb{N}$ are omitted. We present some tools to circumvent these difficulties.

5.2.1 Computation of the residuals

Let us consider the k-th step of the Kleinman-Newton method (3.2) utilizing the ADI method as iterative solver. For each ADI iterate $X_{k+1,i}$ the resulting residual is defined as

$$R_k^{(i)} = X_{k+1,i}A_k + A_k^T X_{k+1,i} + S_k, \tag{5.6}$$

which can not be computed without $X_{k+1,i}$. Since the feedback algorithms only provide $B^T X_{k+1,i}$, we need a reformulation of the residual equation.
Banks and Ito [3] give an explicit formula for the residual for the factored Smith method. We already presented an equivalent formula for the ADI method in Lemma 3.5.2, namely

$$R_k^{(i)} = A_{k,\mu_i}^T ... A_{k,\mu_1}^T (X_{k+1,0}A_k + A_k^T X_{k+1,0} + S_k) A_{k,\mu_1} ... A_{k,\mu_i}.$$

If we choose $X_{k+1,0}$ as zero matrix we will obtain an equation for the residual of the i-th ADI iterate in the k-th Newton step. The computation of A_k, $A_{k,\mu_i},\ i \in \mathbb{N}$ and S_k does not require the knowledge of $X_{k+1,i}$ only $B^T X_{k+1,i}$ is necessary, since

$$\begin{aligned} A_k &= A - BB^T X_k \\ A_{k,\mu_i} &= (A_k - \mu_i I)(A_k + \mu_i I)^{-1} \\ S_k &= X_k BB^T X_k + C^T C. \end{aligned}$$

As a result we obtain a representation of the residuals, which can be computed within a feedback framework.
Note, low-rank ADI methods are mathematically equivalent to the ADI method. Therefore the same representation of the residuals can be utilized if any variant of the low-rank ADI methods, presented in chapter 3.4, is applied for the solution of the occurring Newton steps.

5.2.2 Calculation of $\mathcal{F}(X_k)$

In addition, the computation of $\mathcal{F}(X_k)$ is not a serious problem as long the initial iterate $X_0 = 0$ and therefore $B^T X_0 = 0$ is chosen. We obtain $\mathcal{F}(X_0) = C^T C$ with the definition of the algebraic Riccati equation (5.2).
The calculation of $\mathcal{F}(X_{k+1})$ is possible for all $k \geq 0$ due to the quadratic nature of the mapping

$$\begin{aligned}\mathcal{F}(X_{k+1}) &= \mathcal{F}(X_k) + \mathcal{F}'(X_k)(X_{k+1} - X_k) + \frac{1}{2}\mathcal{F}''(X_k)(X_{k+1} - X_k, X_{k+1} - X_k) \\ &= R_k - (X_{k+1} - X_k)^T B B^T (X_{k+1} - X_k) \end{aligned} \tag{5.7}$$

where the inexact Newton step (2.7) was exploited. Remember that $B^T X_k$ is known for all $k \in \mathbb{N}$.
After these two remarks we are able to compute $\mathcal{F}(X_k)$ and R_k without knowledge of the Newton iterates X_k as long as we define $X_{k,0} = 0$ for all $k \geq 0$. The choice of the zero matrix as initial iterate for the iterative solver is a considerable restriction but it is often common practice.
Now Theorem 2.2.1 gives a stringent guidelines to terminate the inner iteration of an inexact feedback gain algorithm. By varying the accuracy of the Newton steps, we obtain inexact methods with a linear, superlinear or even quadratic rate of convergence.
Of course, the additional numerical effort to compute the residuals is significant. Alternatively one could use heuristic stopping criteria depending on relative changes of the feedback matrices as presented and implemented in the LyaPack package [58].

Chapter 6

General convergence theory

Newton's method is regarded as one of the most powerful tools in solving nonlinear equations. Hence it is very well analyzed and various extensions are presented in the literature, e.g. quasi Newton methods including DFP, BFGS, SR1 updates [20, 24, 15, 23, 28, 68, 17], globalizations techniques via line search [30] and many more.
Sometimes an interesting characteristic about Newton's method can be observed in a monotone convergence behavior. Both types of monotonicity, increasing and decreasing, can be found in several applications [45, 19, 25]. A theoretical background to explain these phenomena have been already stated e.g. in [56, 61].
But in case of inexact Newton's methods these results are no longer valid. The possibility to extend similar monotonicity results to inexact Newton's methods is indicated by Theorem 2.3.4. Here we were able to restore the monotone convergence property of Newton's method applied to the algebraic Riccati equation (1.1) also for an inexact Newton version. Our goal is now to introduce a new theory to describe this behavior in a more general framework.
In the next section, we introduce some relevant theoretical aspects including regular proper convex cones [73] and inverse positive [negative] mappings. Chapter 6.2 presents a convergence theory for concave mappings, which secures a monotone convergence of the inexact Newton method. Of course, this theory can be also extended to convex mappings. The main results are stated for the convex case in the final section of this chapter.

6.1 Theoretical background

Throughout this section, we tackle the question under which requirements on the mapping $\mathcal{F}$ and the residuals R_k an inexact version of Newton's method

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) = -\mathcal{F}(X_k) + R_k \tag{6.1}$$

exhibits a monotone decreasing [or increasing] convergence property, i.e. $X_k \geq X_{k+1}$ $[X_k \leq X_{k+1}]$. Here we always consider $\mathcal{F} : D \subset E \to F$, where E and F are Banach spaces, D is an open convex subset of E.
Since we are interested in the monotonicity of the inexact Newton iterates, we have to introduce at first on both Banach spaces a partial ordering. This can be done with help of a proper convex cone K_E respectively K_F.

Definition 6.1.1. *A proper convex cone is a subset K of a Banach space such that $K + K \subset K$, $\alpha K \subset K$ for all $\alpha > 0$, and $K \cap -K = \{0\}$.*

Remark 6.1.2. *With help of a proper convex cone K, we are able to introduce a partial ordering on the Banach space and call $x \leq y$ if and only if $y - x \in K$.*

All convergence results, presented in section 6.2 and section 6.3, are based on a specific quality of the Banach space E. It is required, that the monotonicity and boundedness of a sequence $\{x_k\}_{k\in\mathbb{N}} \in E^{\mathbb{N}}$ induce its convergence. For some spaces this is trivial fact but not for general Banach spaces.
Therefore we assume the cone K_E to be regular [73, 38], which secures the convergence of a monotone and bounded sequence.

Definition 6.1.3. *Let K be a proper convex cone on a Banach space E. The cone K is called regular if every increasing (or decreasing) sequence $\{x_k\}_{k\in\mathbb{N}} \in E^{\mathbb{N}}$, which is order-bounded from above (below), is convergent.*

The concept of regular cones seems very plausible. Nevertheless, it is not a trivial task to equip a Banach space with a regular cone, especially in the case of function spaces. We present some examples and extensions of regular cones for a better understanding.
Regular cones can be easily introduced on the n-dimensional Euclidian space $\mathbb{R}^n$ and on $\mathcal{L}^p([0,1])$ spaces for $0 < p < \infty$.

Example 6.1.4. *The subset $K_1 := \{x = (x_1, ..., x_n) \in \mathbb{R}^n |\ x_i \geq 0, \quad 1 \leq i \leq n\}$ defines a regular cone on the n-dimensional Euclidian space.*

This subset K_1 is the obvious choice of a cone on the n-dimensional Euclidian space and satisfies all characteristics of a proper convex cone (Definition 6.1.1).

The Bolzano-Weierstrass theorem together with the monotonicity of the iterates secure the convergence of every increasing (or decreasing) and order-bounded sequence, which defines the regularity of the cone K_1.
For some function spaces there also exists a natural choice of a proper convex cone.

Example 6.1.5. *On $\mathcal{L}^p([0,1])$ spaces, where $0 < p < \infty$, the subset $K_2 = \{f \in \mathcal{L}^p([0,1])|\ f(t) \geq 0 \ \ a.e.\}$ defines a proper convex cone, which is regular.*

For a definition of $\mathcal{L}^p([0,1])$ spaces see e.g. [65]. K_2 obviously satisfies all characteristics of a proper convex cone. Example 2.2.1 in [38] demonstrates the regularity of K_2 in case of $\mathcal{L}^p([0,1])$ spaces with $0 < p < \infty$.
On both spaces the most obvious choice of a proper convex convex cone leads to a regular cone. But on other function spaces it is much harder to obtain a regular cone.

Example 6.1.6. *Consider $C[0,1]$, the space of real-valued continuous functions, defined on $[0,1]$. Here the subset $K_3 = \{f \in C[0,1]|\ f(t) \geq 0,\ t \in [0,1]\}$ is a proper convex cone, which is not regular.*

In order to prove the non-regularity of the cone K_3, we present an example taken from [73]: $x_n(t) = -t^n$ defines a sequence in $C[0,1]$. This sequence shows a monotone behavior, i.e. $x_1 \leq x_2 \leq ... \leq \hat{x}$ and is bounded from above with $\hat{x}(t) \equiv 0$. It is well known that $\{x_n\}_{n\in\mathbb{N}}$ converges point-wise but not uniformly against

$$\lim_{n\to\infty} x_n(t) = \begin{cases} 0, & t \in [0,1) \\ -1, & t = 1. \end{cases}$$

However, in $C[0,1]$ exists no limit of the sequence $\{x_n\}_{n\in\mathbb{N}}$.

Remark 6.1.7. *The same example can be used to show that $K_4 = \{f \in C^n[0,1]|\ f(t) \geq 0,\ t \in [0,1]\}$ is not a regular cone in $C^n[0,1]$, which describes the space of n- times differentiable real-valued functions, defined on $[0,1]$.*

All above-mentioned examples of proper convex cones describe the natural choice of a cone. Regular cones, which are not trivial, can be introduced for example on $\mathbb{R}^{n\times n}$, the space of real quadratic $n \times n$ matrices.

Example 6.1.8. *The subset $K_5 = \{A \in \mathbb{R}^{n\times n}|\ A$ is non-negative definite$\}$ defines a regular proper convex cone on $\mathbb{R}^{n\times n}$.*

Obviously, all requirements of a proper convex cone (Definition 6.1.1) are satisfied. Without loss of generality, we restrict ourselves to the case of monotone increasing and bounded sequences $\{X_n\}_{n\in\mathbb{N}}$ to testify the regularity of K_5, i.e. $X_1 \leq ... \leq X_k \leq X_{k+1} \leq ... \leq \hat{X}$, where $X_k = (x_{ij}^k)$. Since every monotone and bounded sequence of real numbers is convergent, we obtain the existence of $\lim_{k\to\infty} a^T X_k a$ for every $a \in \mathbb{R}^n$. If we choose $a = e_i$, the i- th unit vector, we will achieve the convergence of the diagonal entries $\{x_{ii}^k\}_{k\in\mathbb{N}}$ for $i = 1, .., n$. Due to the choice $a = e_i + e_j$ and the symmetry of X_k, we get the convergence of $a^T X_k a = x_{ii}^k + 2x_{ij}^k + x_{jj}^k$. Given the convergence of the diagonal entries, we obtain the convergence of all other entries. Therefore the existence of $\lim_{k\to\infty} X_k$ is guaranteed, which states the regularity of K_5.
In some cases there exists no possibility to establish a regular cone in a space, even though one requires to infer convergence of a sequence from its monotonicity and boundedness. Burns, Sachs and Zietsman [16] were able to avoid this problem in case of infinite dimensional Riccati equation in Hilbert spaces, where a sequence of self-adjoint operators occurred.

Example 6.1.9. *$\mathcal{L}(H)$ denotes the Banach space of linear bounded operators from one Hilbert space H into H and $\Sigma(H) := \{\Pi \in \mathcal{L}(H) |\ \Pi$ is self-adjoint$\}$. We call a self-adjoint operator A positive, $A > 0$, if*

$$\langle Ax, x\rangle \geq 0$$

holds for all $x \in H$.

Here $\mathcal{L}(H)$ defines a proper convex cone and a partial ordering can be introduced and we call $A \geq B$ if and only if $A - B$ is positive. Instead of discussing the regularity of $\mathcal{L}(H)$, we present an interesting Theorem, taken from [53, p. 282].

Theorem 6.1.10. *If $\{A_n\}$ is a sequence of self-adjoint mutually commutative operators, if $A_n B = B A_n$ for all n and if*

$$A_1 \leq ... \leq A_n \leq A_{n+1} \leq ... \leq B,$$

then A_n converge to A, and $A \leq B$. (An analogous statement holds for monotone decreasing sequences).

Under additional requirements on the sequence A_n, Theorem 6.1.10 enables us to conclude convergence from monotonicity and boundedness. This can be seen as an extension of the concept of regular cones in case of self-adjoint operators. However, in the context of this thesis, following assumption generates a sufficient theoretical background.

Assumption 6.1.11. *K_E and K_F are assumed to be proper convex cones. In addition, the cone K_E is needed to be regular.*

Only for certain classes of mappings $\mathcal{F}$, we are able to state a monotonicity preserving convergence theory for the inexact Newton's methods.
On the one hand we require that $\mathcal{F}$ has a Fréchet derivative for all X in D and a concept of inverse positivity (negativity) plays an important role.

Definition 6.1.12. *Let E and F be Banach spaces and $\mathcal{F} : D \subset E \to F$, where D is a open subset of E. The mapping $\mathcal{F}$ is called Fréchet differentiable at $x \in D$ if there exists a bounded linear operator $A_x : E \to F$ such that*

$$\lim_{h \to 0} \frac{\|\mathcal{F}(x+h) - \mathcal{F}(x) - A_x(h)\|_F}{\|h\|_E} = 0.$$

See e.g. [65] for details on Fréchet derivatives.

Definition 6.1.13. *Let $\mathcal{L} : D \subset E \to F$ be a linear mapping and $Z \in D$. $\mathcal{L}$ is called inverse positive [negative] if $\mathcal{L}^{-1}$ exists and $\mathcal{L}(Z) \geq 0$ implies $Z \geq 0$ [$Z \leq 0$].*

On the other hand we need a special structure of the mapping, i.e. $\mathcal{F}$ is required to be a concave or convex mapping.
In the next section, we will introduce in detail a monotonicity preserving convergence theory for inexact Newton's methods in case of concave mappings. These results can be easily transferred into the convex context. For sake of completeness the main results are presented in chapter 6.3 also for the convex case.

6.2 Concave theory

Throughout this section we consider a concave mapping $\mathcal{F}$.

Definition 6.2.1. *Let E and F be Banach spaces and assume that D is an open convex subset of E. We call a Fréchet differentiable mapping $\mathcal{F} : D \subset E \to F$ concave on D if and only if*

$$\mathcal{F}'(X)(Y - X) \geq \mathcal{F}(Y) - \mathcal{F}(X) \tag{6.2}$$

holds for all $X, Y \in D$.

In order to design a convergence theory, we establish several statements:
By introducing a solution of the inequality $\mathcal{F}(X) \geq 0$ we are able to prove the boundedness of the iterates.

Corollary 6.2.2. *Let $\hat{X} \in D$ satisfy $\mathcal{F}(\hat{X}) \geq 0$, $\mathcal{F}(\cdot)$ be concave on D, $\mathcal{F}'(X_k)$ be inverse negative [positive] and $R_k \leq \mathcal{F}(\hat{X})$. Under these requirements the next inexact Newton iterate X_{k+1}, defined by (6.1), satisfies $X_{k+1} \geq \hat{X}$ $[X_{k+1} \leq \hat{X}]$.*

Proof. The inexact Newton step (6.1) is equivalent to

$$\mathcal{F}'(X_k)(\hat{X} - X_{k+1}) = \mathcal{F}'(X_k)(\hat{X} - X_k) + \mathcal{F}(X_k) - R_k \tag{6.3}$$

and due to the concavity of $\mathcal{F}$ and the requirements on R_k, we can state

$$\mathcal{F}'(X_k)(\hat{X} - X_{k+1}) \geq \mathcal{F}(\hat{X}) - R_k \geq 0. \tag{6.4}$$

The inverse negativity [positivity] of $\mathcal{F}'(X_k)$ secures $\hat{X} \leq X_{k+1}$ $[\hat{X} \geq X_{k+1}]$. □

In addition, the concavity of the mapping provides an upper bound for the function value.

Corollary 6.2.3. *Let $\mathcal{F}(\cdot)$ be concave on D and assume that X_{k+1} has been determined via a step of the inexact Newton's method (6.1) with residual R_k. Then $\mathcal{F}(X_{k+1}) \leq R_k$.*

Proof. Consider the concavity condition of $\mathcal{F}$

$$\mathcal{F}(X_{k+1}) - \mathcal{F}(X_k) \leq \mathcal{F}'(X_k)(X_{k+1} - X_k).$$

Together with the inexact Newton step (6.1) we obtain

$$\mathcal{F}(X_{k+1}) \leq R_k. \tag{6.5}$$

□

The following Corollary secures the monotonicity of the inexact Newton iterates.

Corollary 6.2.4. *Assume that $\mathcal{F}'(X_k)$ is inverse negative [positive] and $\mathcal{F}(X_k) \leq R_k$. Under these requirements the next inexact Newton iterate X_{k+1} fulfills $X_k \geq X_{k+1}$ $[X_k \leq X_{k+1}]$.*

Proof. The inexact Newton step is

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) = -\mathcal{F}(X_k) + R_k \geq 0 \tag{6.6}$$

which implies $X_k \geq X_{k+1}$ $[X_k \leq X_{k+1}]$ due to the inverse negativity [positivity] of $\mathcal{F}'(X_k)$. □

We can combine the statements of Corollary 6.2.2 and Corollary 6.2.4 to prove a first convergence Theorem.

Theorem 6.2.5. *Let $\hat{X} \in D$ satisfy $\mathcal{F}(\hat{X}) \geq 0$, $\mathcal{F}(\cdot)$ be concave on D, $\mathcal{F}'(X_k)$ be inverse negative [positive] for all iterates $X_k,\ k \in \mathbb{N}_0$ and $R_0 \leq \mathcal{F}(\hat{X})$. Under following requirements on the residuals*

$$\mathcal{F}(X_k) \leq R_k \leq \mathcal{F}(\hat{X}) \qquad k = 1, ... \tag{6.7}$$

$$\lim_{k\to\infty} R_k = 0 \tag{6.8}$$

we obtain monotone iterates $X_1 \geq ... \geq X_k \geq X_{k+1} \geq ... \geq \hat{X}$ $[X_1 \leq ... \leq X_k \leq X_{k+1} \leq ... \leq \hat{X}]$ for the inexact Newton's method (6.1), which converge to a solution of the equation $\mathcal{F}(X) = 0$.

Proof. Corollary 6.2.2 states the boundedness of the sequence $(X_k)_{k\in\mathbb{N}}$ and Corollary 6.2.4 proves the monotonicity of the iterates. Since K_E is assumed to be a regular cone, the inexact Newton iterates are convergent to some limiting matrix X_∞. Equation (6.1) and $\lim_{k\to\infty} R_k = 0$ prove $\mathcal{F}(X_\infty) = 0$. □

Remark 6.2.6. *i) Compared to standard inexact Newton statements, e.g. Theorem 2.2.1, Theorem 6.2.5 provides a more global convergence result. The initial iterate does not have to be close to a solution.*
ii) If, in addition, $F(X_0) \leq R_0$ is satisfied, the monotonicity will include the initial iterate X_0 as well, i.e. $X_0 \geq X_1 \geq ...[X_0 \leq X_1 \leq ...]$. This is a direct result of Corollary 6.2.4.
iii) The requirements on R_k are obviously satisfied for exact Newton methods due to Corollary 6.2.3 with $R_k = 0$.

We can also state an alternative version of the developped convergence theory utilizing Corollary 6.2.3.

Theorem 6.2.7. *Let $\hat{X} \in D$ satisfy $\mathcal{F}(\hat{X}) \geq 0$, $\mathcal{F}(\cdot)$ be concave on D, $\mathcal{F}'(X_k)$ be inverse negative [positive] for all iterates $X_k,\ k \in \mathbb{N}_0$. Under following requirements on the residuals*

$$R_k \leq \mathcal{F}(\hat{X}) \qquad k = 0, ... \tag{6.9}$$

$$R_k \leq R_{k+1} \qquad k = 0, ... \tag{6.10}$$

$$\lim_{k\to\infty} R_k = 0 \tag{6.11}$$

we obtain monotone iterates $X_1 \geq ... \geq X_k \geq X_{k+1} \geq ... \geq \hat{X}$ $[X_1 \leq ... \leq X_k \leq X_{k+1} \leq ... \leq \hat{X}]$ for the inexact Newton's method (6.1), which converge to a solution X_∞ of the equation $\mathcal{F}(X) = 0$.

Proof. Accordingly to Corollary 6.2.2, the requirements $R_k \leq \mathcal{F}(\hat{X})$ for all $k \in \mathbb{N}_0$ secure the boundedness of the inexact Newton iterates. Corollary 6.2.3 proves $\mathcal{F}(X_k) \leq R_{k-1}$ for every $k \in \mathbb{N}$ and requirement (6.10) leads to $\mathcal{F}(X_k) \leq R_{k-1} \leq R_k$. With help of Corollary 6.2.4, we see that the iterates are monotone and as a result, we obtain the convergence of the sequence $\{X_k\}$ to a solution similar to Theorem 6.2.5. □

Remark 6.2.8. *i) The requirement $R_k \leq R_{k+1}$ of latter theorem looks unusual but the residuals are allowed to be negative.*
ii) Requirements of Theorem 6.2.7 are more restrictive compared to those introduced in Theorem 6.2.5. In Theorem 6.2.5 the residuals are allowed to be positive but (6.10) together with (6.11) can be only fulfilled for negative R_k.

In summary, convergence theorems for inexact Newton's methods have been presented, which provide a monotone convergence behavior for inexact Newton iterates. The theory is applicable as long as three main requirements on the mapping $\mathcal{F}$ are satisfied:
Concave systems show a monotone convergence behavior. But we do not need the concavity condition (6.2) for all $X, Y \in D$. As long as (6.2) is fulfilled for all points needed in Corollary 6.2.2 and Corollary 6.2.3, all proofs can be completed. An extension to convex mappings will be presented in section 6.3.
In order to prove the boundedness of the iterates, a solution of the inequality $\mathcal{F}(X) \geq 0$ is necessary.
Finally the derivative $\mathcal{F}'(X_k)$ has to provide inverse negativity [positivity] in all inexact Newton iterates $X_k,\ k \in \mathbb{N}_0$.
These three conditions are restrictive, but several important and well known problems match this theory and will be presented in chapter 7.
Convergence rates can not be stated within the above mentioned theory. Additional knowledge on the considered mappings is necessary to provide rate estimates. We will show some rate estimates for the examples in section 7.

6.3 Convex theory

In chapter 6.2 we introduced a monotonicity preserving inexact Newton's method for concave systems. Of course all results and statements of the this chapter can be adjusted also for convex mappings.

Definition 6.3.1. *Let E and F be Banach spaces and assume that D is a open convex subset of E. We call a Fréchet differentiable mapping $\mathcal{F} : D \subset E \to F$ convex on D if and only if*

$$\mathcal{F}'(X)(Y - X) \leq \mathcal{F}(Y) - \mathcal{F}(X) \tag{6.12}$$

holds for all $X, Y \in D$.

Remark 6.3.2. *It is obvious that the convexity of a mapping* $\mathcal{F}$ *implies the concavity of* $-\mathcal{F}$.

For a convex mapping $\mathcal{F}$ we can apply an inexact Newton's method to the concave system $-\mathcal{F}(X) = 0$ and utilize all theory of section 6.2 and state similar convergence results for the convex case.
We do not want to repeat every Corollary, only the equivalences of the two main convergence results are mentioned.

Theorem 6.3.3. *Let* $\mathcal{F}(\cdot)$ *be convex on* D, $\hat{X} \in D$ *satisfy* $\mathcal{F}(\hat{X}) \leq 0$, $\mathcal{F}'(X_k)$ *be inverse positive [negative] for all iterates* X_k, $k \in \mathbb{N}_0$ *and* $R_0 \leq -\mathcal{F}(\hat{X})$. *Under following requirements on the residuals*

$$-\mathcal{F}(X_k) \leq R_k \leq -\mathcal{F}(\hat{X}) \qquad k = 1, ... \tag{6.13}$$

$$\lim_{k\to\infty} R_k = 0 \tag{6.14}$$

we obtain monotone iterates $X_1 \geq ... \geq X_k \geq X_{k+1} \geq ... \geq \hat{X}$ $[X_1 \leq ... \leq X_k \leq X_{k+1} \leq ... \leq \hat{X}]$ *for the inexact Newton's method (6.1) which converge to a solution of the equation* $\mathcal{F}(X) = 0$.

Proof. The proof follows directly with Theorem 6.2.5, applied to the concave mapping $-\mathcal{F}$. □

Analogous to Theorem 6.2.7 we can state a second convergence result, utilizing the monotonicity of the residuals.

Theorem 6.3.4. *Let* $\hat{X} \in D$ *satisfy* $\mathcal{F}(\hat{X}) \leq 0$, $\mathcal{F}(\cdot)$ *be convex on* D, $\mathcal{F}'(X_k)$ *be inverse positive [negative] for all iterates* X_k, $k \in \mathbb{N}_0$. *Under following requirements on the residuals*

$$R_k \leq -\mathcal{F}(\hat{X}) \qquad k = 0, ...$$

$$R_k \leq R_{k+1} \qquad k = 0, ...$$

$$\lim_{k\to\infty} R_k = 0$$

we obtain monotone iterates $X_1 \geq ... \geq X_k \geq X_{k+1} \geq ... \geq \hat{X}$ $[X_1 \leq ... \leq X_k \leq X_{k+1} \leq ... \leq \hat{X}]$ *for the inexact Newton's method (6.1) which converge to a solution* X_∞ *of the equation* $\mathcal{F}(X) = 0$.

Proof. Again, the concavity of $-\mathcal{F}$ together with Theorem 6.2.7 secures the statements of the Theorem. □

As a result, we easily transferred all results of the concave case to the convex situation.

Chapter 7

Applications

In the previous chapter we introduced a convergence theory for inexact Newton methods including monotonicity of the iterates. Our restrictions on the considered mappings and spaces are restrictive, e.g. we require inverse negativity [positivity] or concavity [convexity] of the mappings. Nevertheless we find many examples from various applications which fit to our assumptions and show the benefits of our theory.
A first example can be found in the algebraic Riccati equation (1.1). We already discussed inexact Kleinman-Newton methods and convergence results for this case in chapter 2.
In section 7.1 we analyse the nonsymmetric algebraic Riccati equation (NARE) which plays an important role in transport theory [42, 43]. Newton‘s method is usually used to solve this equation [34, 12], an inexact version utilizing a doubling iteration scheme has been recently presented in [27]. This equation is closely related the symmetric case but a different theoretical background is relevant, nevertheless is our theory applicable.
Damm and Hinrichsen [19] applied Newton's method for an abstract rational matrix equation, which is discussed in section 7.2. This equation is not only of theoretical interest, it covers the continuous algebraic Riccati equation (CARE) [48], the discrete algebraic Riccati equation (DARE) [48], matrix equations occurring in stochastic control [76] and disturbance attenuation problems [40, 14, 13] as special cases. We established an inexact variant of Newton's method and proved convergence rates.
In a last step, we analyzed the quasilinearization technique, introduced by Bellman and Kalaba [4]. Here one tries to write a parabolic PDE as a solution of a corresponding equation [63]. We analyze the PDE and a discretized version of the PDE in chapter 7.3

7.1 Nonsymmetric algebraic Riccati equation

Our first example plays an important role in applications of transport theory [42, 43] and has been intensively studied in recent years, see e.g. [34, 12, 27].

Definition 7.1.1. *The nonsymmetric algebraic Riccati equation (NARE) is defined as*

$$\mathcal{F}(X) = XCX - XD - AX + B = 0, \tag{7.1}$$

where $\mathcal{F} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ *and* A, B, C, D *are real matrices of sizes* $m\times m$, $m\times n$, $n\times m$ *respectively* $n\times n$.

Even though above nonsymmetric algebraic Riccati equation bears a striking similarity to the algebraic Riccati equation (1.1), discussed in chapter 2, a different theoretical background is relevant. On the one hand in the NARE context we are only interested in the minimal non-negative solutions of $\mathcal{F}(X) = 0$, where a different partial ordering is introduced on $\mathbb{R}^{m\times n}$. On the other hand the NARE is a convex mapping, whereas the ARE is concave..
We shortly review some relevant definitions and statements, which are important for the NARE case. On $\mathbb{R}^{m\times n}$ we introduce a partial ordering with help of following proper convex cone.

Definition 7.1.2. $K = \{A \in \mathbb{R}^{m\times n} |\ a_{ij} \geq 0,\ \ 1 \leq i \leq m,\ 1 \leq j \leq n\}$ *defines a proper convex cone on* $\mathbb{R}^{m\times n}$, *which is obviously regular.*

Remark 7.1.3. *The cone, introduced in Definition 7.1.2, is closely related to the regular cone introduced on the n- dimensional Euclidian space, see Example 6.1.4.*

For $A, B \in \mathbb{R}^{m\times n}$ we therefore call $A < B$ $[A \leq B]$ if and only if $a_{ij} < b_{ij}$ $[a_{ij} \leq b_{ij}]$ and A non-negative [non-positive] if $a_{ij} \geq 0$ $[a_{ij} \leq 0]$ for all i, j.
In order the secure the existence of a non-negative solution of (7.1), we introduce and utilize the concept of M- matrices.

Definition 7.1.4. *A real square matrix* A *is called a* Z*-matrix if all its off-diagonal elements are non-positive and we can write* $A = sI - B$ *with* $B \geq 0$. *A* Z*-matrix is called an* M*- matrix if* $s \geq \rho(B)$, *where* $\rho(\cdot)$ *is the spectral radius.*

Together with the system matrix M, defined below, we are able to state a existence theory for a minimal non-negative solution of the NARE.

Assumption 7.1.5. *We assume throughout this section that*

$$M := \begin{bmatrix} D & -C \\ -B & A \end{bmatrix} \in \mathbb{R}^{m+n,m+n} \tag{7.2}$$

is a nonsingular M-matrix.

Remark 7.1.6. *i) Theorem 3.1 in [32] proves that Assumption 7.1.5 secures the existence of a minimal non-negative solution of (7.1).*
ii) In addition, the M-matrix property leads to $B, C \geq 0$ and we know that A and D are as well nonsingular M-Matrices. Therefore the matrix $I \otimes A + D^T \otimes I$ is also a nonsingular M-matrix, where $\otimes$ is the Kronecker product. See Remark 1.1 in [37] for details.
iii) Let S be the minimal non-negative solution of (7.1). Theorem 2.5 in [33] proves the M-matrix properties of $M_S := I \otimes (A - SC) + (D - CS)^T \otimes I$.

Newton's method has been successfully applied for the solution of the NARE, i.e. $\mathcal{F}(X) = 0$, where a monotone convergence of the resulting Newton iterates was observed [37, 34]. Comparing these results to the general convergence theory of section 6.3, we hope to identify the NARE as an interesting application of our theory. As a result we will be able to establish convergence results, including monotonicity, also in the case of inexact Newton's methods for the solution of the NARE.
Corresponding to chapter 6.3, we analyze the structure of the NARE $\mathcal{F}$ and test the applicability of our main convergence theorems. At first we state a result concerning the convexity of $\mathcal{F}$.

Theorem 7.1.7. *The map $\mathcal{F}$ is convex in following sense:*

$$\mathcal{F}'(X)(Y - X) \leq \mathcal{F}(Y) - \mathcal{F}(X) \qquad \forall\ X, Y \in \mathbb{R}^{m \times n} \text{ with } Y - X \geq 0 \tag{7.3}$$

Proof. The proof follows directly with the Taylor expansion of the nonsymmetric algebraic Riccati equation

$$\mathcal{F}(Y) = \mathcal{F}(X) + \mathcal{F}'(X)(Y - X) + \frac{1}{2}\mathcal{F}''(X)(Y - X, Y - X) \tag{7.4}$$

where the quadratic term

$$\frac{1}{2}\mathcal{F}''(Z)(W, W) = WCW^T \geq 0 \quad \text{for all } W \geq 0 \tag{7.5}$$

due to the non-negativity of C. □

With Assumption 7.1.5 we can state another result, namely the existence of a solution of the inequality $\mathcal{F}(X) \leq 0$. Let S be the minimal non-negative solution of (7.1), i.e. $\mathcal{F}(S) = 0$ and of course S is also a solution of $\mathcal{F}(X) \leq 0$.
Finally, we have to consider the derivative of the mapping $\mathcal{F}$ and analyze the inverse negativity of $\mathcal{F}'(X_k)$. In this case two alternative presentations of the derivate can be developed.

Corollary 7.1.8. *The Fréchet derivative of $\mathcal{F}$ at a given matrix X is a linear map $\mathcal{F}'(X) : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ and is defined as*

$$\mathcal{F}'(X)(Z) = -((A - XC)Z + Z(D - CX)). \tag{7.6}$$

Corollary 7.1.9. *With help of the Kronecker product we are able to rewrite $\mathcal{F}'(X)Z$ as matrix-vector product*

$$-(I \otimes (A - XC) + (D - CX)^T \otimes I)vec(Z) =: -M_X vec(Z) \tag{7.7}$$

where vec stacks the columns of the matrix Z into a long vector.

Remark 7.1.10. *Therefore we can identify $\mathcal{F}'(X)$ with the matrix $-M_X$.*

The inverse negativity can now be proved with help of (7.7) and a characteristic of M-matrices.

Theorem 7.1.11. *For a Z-matrix $A \in \mathbb{R}^{n\times n}$, the following are equivalent:*

i) A is a nonsingular M-matrix

ii) the linear map defined by A is inverse positive, i.e. $A^{-1} \geq 0$

iii) $A\nu > 0$ for some vector $\nu \in \mathbb{R}^n$ with $\nu > 0$

Proof. For a proof see e.g. [11]. □

This Theorem in combination with Corollary 7.1.9 states the inverse negativity of $\mathcal{F}'(X_k)$ as long as the matrix $I \otimes (A - X_k C) + (D - CX_k)^T \otimes I$ is a nonsingular M-matrix.
For the zero matrix as the initial iterate for our inexact Newton's method, i.e. $X_0 = 0 \in \mathbb{R}^{m\times n}$ we can prove an easy result.

Corollary 7.1.12. *If $X_0 = 0 \in \mathbb{R}^{m\times n}$ then $\mathcal{F}'(X_0)$ is inverse negative.*

Proof. Instead of showing $\mathcal{F}'(X_0)Z \geq 0 \Rightarrow Z \leq 0$ we utilize the matrix-vector product (7.7) with $X_0 = 0 \in \mathbb{R}^{m\times n}$:

$$M_{X_0}vec(Z) = -(I \otimes A + D^T \otimes I)vec(Z) \tag{7.8}$$

With Assumption 7.1.5 and Remark 7.1.6 ii) we obtain the M- matrix property of $I \otimes A + D^T \otimes I$ and with Theorem 7.1.11 the inverse negativity of $\mathcal{F}'(X_0)$. □

To prove the inverse negativity in all inexact Newton iterates, another result on M-matrices is necessary.

Lemma 7.1.13. *Let A be a nonsingular M-matrix. If $B \geq A$ is a Z-matrix, then B is also a nonsingular M-matrix.*

Proof. The equivalence of i) and iii) in Theorem 7.1.11 yields the proof. □

Lemma 7.1.13 enables us to testify the inverse negativity of $\mathcal{F}'(X_k)$ for all inexact Newton iterates X_k via induction.

Theorem 7.1.14. *Let $X_0 = 0 \in \mathbb{R}^{m\times n}$, S be the minimal non-negative solution of (7.1) and $-\mathcal{F}(X_k) \leq R_k \leq -\mathcal{F}(S) = 0$ for all $k \in \mathbb{N}_0$. The inexact Newton iterates X_k, defined in (6.1), lead to inverse negative $\mathcal{F}'(X_k)$.*

Proof. The proof is via induction. Corollary 7.1.12 states the result for $k = 0$, which is the induction hypothesis. Now we assume that $\mathcal{F}'(X_k)$ is inverse negative for $k = 0, ..., i$. On the one hand $R_i \leq 0$ and $\mathcal{F}'(X_i)$ is inverse negative, as a result we achieve $X_{i+1} \leq S$, due to Corollary 6.2.2, applied to the concave mapping $-\mathcal{F}$. On the other hand $-\mathcal{F}(X_i) \leq R_i$ and Corollary 6.2.4 secures $0 \leq X_i \leq X_{i+1}$. We consider the matrix representation $M_{X_{i+1}}$ of $-\mathcal{F}'(X_{i+1})$ in Remark 7.1.6 iii)

$$M_{X_{i+1}} = I \otimes (A - X_{i+1}C) + (D - CX_{i+1})^T \otimes I$$

respectively M_S of $-\mathcal{F}'(S)$ in (7.7)

$$M_S = I \otimes (A - SC) + (D - CS)^T \otimes I.$$

With respect to Definition 7.1.4 we can state the Z- matrix property of $M_{X_{i+1}}$ because A, D are M-matrices and $X_{i+1}C$, CX_{i+1} are both non-negative. Since $X_{i+1} \leq S$, we can utilize Lemma 7.1.13 to obtain the M-matrix property of $M_{X_{i+1}}$ and as a result the inverse negativity of $\mathcal{F}'(X_{i+1})$, which concludes the induction. □

All requirements of the monotonicity preserving theory for inexact Newton's methods can be satisfied and we can state a convergence result for the non-symmetric algebraic Riccati equation:

Theorem 7.1.15. *Let $X_0 = 0 \in \mathbb{R}^{m\times n}$, and S be the minimal non-negative solution of (7.1). Under following requirements on the residuals*

$$-\mathcal{F}(X_k) \leq R_k \leq 0 \qquad k = 0, ... \tag{7.9}$$

$$\lim_{k\to\infty} R_k = 0 \tag{7.10}$$

we obtain monotone iterates $X_0 \leq X_1 \leq ... \leq X_k \leq X_{k+1} \leq ... \leq S$ for the inexact Newton's method (6.1), which converge to the minimal solution of the equation $\mathcal{F}(X) = 0$.

Proof. Theorem 7.1.7 states the convexity of $\mathcal{F}$ in all relevant points. Due to Assumption 7.1.5 the existence of a minimal non-negative solution S of (7.1) is guaranteed and we can also see S as a solution of the corresponding inequality $\mathcal{F}(X) \leq 0$. Theorem 7.1.14 testifies that $\mathcal{F}'(X_k)$ is inverse negative for all inexact Newton iterates X_k, assumed the zero matrix as initial iterate. Since all requirements of Theorem 6.3.3 are satisfied, we can transfer all convergence results with help of Theorem 6.3.3. Therefore the convergence of the sequence $\{X_k\}$ to a solution of $\mathcal{F}(X) = 0$ is proven and since $X_k \leq S$ we achieve the convergence to the minimal solution S.

□

As a result we proved the applicability of inexact Newton methods to nonsymmetric Riccati equations corresponding to our theoretical background. The monotonicity of the iterates can be also restored for the inexact case as shown in Theorem 7.1.15.
An inexact variant of Newton's method for NARE has been developed in a recent paper [27]. The authors utilize a doubling iteration scheme to solve their inner iteration.

7.2 General rational matrix equation

Damm and Hinrichsen [19] applied Newton's method for an abstract rational matrix equation, which includes the continuous algebraic Riccati equation (CARE) [48], the discrete algebraic Riccati equation (DARE) [48], matrix equations occurring in stochastic control [76] and disturbance attenuation problems [40, 14, 13] as special cases. More details on rational matrix equations in stochastic control can be found in [18].
This equation can be seen as an extension of the algebraic Riccati equation (1.1), already discussed in chapter 2. For this reason, we hope to establish a monotonicity preserving convergence theory also for inexact Newton's method, applied to

this complicated rational matrix equation. As a result we achieve convergence results for all special cases, mentioned above.
At first some introductory definitions and remarks are necessary.

Definition 7.2.1. *In this section we call $\mathcal{H}^n \subset K^{n\times n}$ the real space of $n \times n$ Hermitian matrices with entries in K and $\mathcal{H}^n_+$ the convex set of non-negative definite matrices.*

Remark 7.2.2. *$\mathcal{H}^n$ is endowed with the Frobenius inner product $\langle X, Y\rangle = trace(XY)$ and with the corresponding norm $\|X\| = \langle X, X\rangle^{\frac{1}{2}}$.*

A partial ordering is introduced with help of a proper convex cone $K = \{A \in \mathcal{H}^n |\ A \ \ is\ non\text{-}negative\ definite\}$, see Example 6.1.8, and we call $A \geq B$ if and only if $A - B$ is non-negative definite. In addition, we call $A > 0$ for positive definite. For a matrix A we call A^* the conjugate transpose of A.
Now all necessary definitions have been introduced and we are able to introduce the rational matrix equation, defined and analyzed by Damm and Hinrichsen [19].

Example 7.2.3. *The equation $\mathcal{F}(X) : dom\ \mathcal{F} := \{X \in \mathcal{H}^n : \mathcal{Q}(X) > 0\} \to \mathcal{H}^n$ is defined as*

$$\mathcal{F}(X) = \mathcal{P}(X) - \mathcal{S}(X)\mathcal{Q}(X)^{-1}\mathcal{S}(X)^*, \tag{7.11}$$

with the affine linear mappings $\mathcal{P} : \mathcal{H}^n \to \mathcal{H}^n$, $\mathcal{Q} : \mathcal{H}^n \to \mathcal{H}^l$ and $\mathcal{S} : \mathcal{H}^n \to K^{n\times l}$

$$\begin{aligned}\mathcal{P}(X) &= A^*X + XA + \Pi_1(X) + P_0\\ \mathcal{S}(X) &= XB + \Sigma(X) + S_0\\ \mathcal{Q}(X) &= \Pi_2(X) + Q_0\end{aligned}$$

where $A \in K^{n\times n}$, $P_0 \in \mathcal{H}^n$, B, $S_0 \in K^{n\times l}$, and $Q_0 \in \mathcal{H}^l$. Additionally $\Pi_1 : \mathcal{H}^n \to \mathcal{H}^n$, $\Pi_2 : \mathcal{H}^n \to \mathcal{H}^l$ and $\Sigma : \mathcal{H}^n \to K^{n\times l}$ are linear mappings and $dom\ \mathcal{F} \neq \emptyset$ is assumed.

Remark 7.2.4. *For theoretical purpose, we need*

$$\Pi : \mathcal{H}^n \to \mathcal{H}^{n+l}, \quad \Pi(X) := \begin{bmatrix} \Pi_1(X) & \Sigma(X) \\ \Sigma(X)^* & \Pi_2(X) \end{bmatrix}$$

to be a positive mapping, i.e. Π maps $\mathcal{H}^n_+$ to $\mathcal{H}^{n+l}_+$.

Damm and Hinrichsen proved a monotone convergence behavior of Newton's method, applied to equation $\mathcal{F}(X) = 0$, see Theorem 6.1 in [19]. We follow closely the notation and theory developed in this paper and therefore we recommend an

extensive study of the original paper.
In order to apply an inexact version of Newton's method which maintains the monotonicity, we check whether above system of equations (7.11) fits to the theory, presented in section 6.2.
The concavity of $\mathcal{F}$ was the first requirement of the monotonicity preserving theory. For this goal we consider a Taylor expansion for $\mathcal{F}$.

Corollary 7.2.5. *Let $X, Y \in dom\ \mathcal{F}$ then following identity holds*

$$\mathcal{F}(Y) = \mathcal{F}(X) + \mathcal{F}'(X)(Y - X) - \Phi_X(Y) \tag{7.12}$$

where $-\Phi_X(Y)$ is the remainder term of the first-order Taylor expansion and $\Phi_X(Y) \geq 0$ holds for all $X, Y \in dom\ \mathcal{F}$.

Proof. For a proof and the definition of $\Phi_X(Y)$, see Proposition 5.5 in [19]. □

As a consequence we can state a first result on the concavity of $\mathcal{F}$.

Corollary 7.2.6. *$\mathcal{F}$ is a concave map on dom $\mathcal{F}$, i.e. for all $X, Y \in dom\ \mathcal{F}$:*

$$\mathcal{F}'(X)(Y - X) \geq \mathcal{F}(Y) - \mathcal{F}(X).$$

Proof. The proof follows directly from Corollary 7.2.5 and the positivity of $\Phi_X(Y)$ for all $X, Y \in dom\ \mathcal{F}$ □

A second condition of our theory was the existence of a solution to inequality $\mathcal{F}(X) \geq 0$, which can be secured under additional requirements for our example.

Assumption 7.2.7. *Let $Q_0 > 0$ and $P_0 \geq S_0 Q_0^{-1} S_0^*$. As a result we achieve $\mathcal{F}(0) = P_0 - S_0 Q_0^{-1} S_0^* \geq 0$.*

In order to prove the last requirement, namely the inverse negativity of the derivative in the inexact Newton iterates, we have to impose some theoretical results.

Definition 7.2.8. *Let $X \in dom\ \mathcal{F}$. $\mathcal{F}'(X)$ is called stable if $\sigma(\mathcal{F}'(X)) \subset \mathbb{C}_- := \{c \in \mathbb{C} | Re\ c < 0\}$, where $\sigma(\mathcal{F}'(X))$ denotes the spectrum of $\mathcal{F}'(X)$.*

Next we realize that the inverse negativity of the derivative is equivalent to the stability of the derivative.

Corollary 7.2.9. *Let $\mathcal{F}'(X)$ be the derivative of $\mathcal{F}$ at $X \in dom\ \mathcal{F}$, then following statements are equivalent:*

i) $\mathcal{F}'(X)$ is stable

ii) $\mathcal{F}'(X)$ *is inverse negative.*

Proof. Proposition 5.2 and Corollary 3.8 in [19] guarantee that $-\mathcal{F}'(X)$ is inverse positive, which leads to the inverse negativity as introduced in Definition 6.1.13. □

Remark 7.2.10. *Above Corollary can be seen as the generalization of Theorem 2.3.2, which was important in case of algebraic Riccati equation.*

With this result, we can apply the theory, developed in section 6.2, under the condition that $\mathcal{F}'(X_k)$ is stable for all inexact Newton iterates. We will introduce a requirement on the residual R_k of the k- th inexact Newton step

$$\mathcal{F}'(X_k)(X_{k+1} - X_k) = -\mathcal{F}(X_k) + R_k \tag{7.13}$$

to secure the stability of $\mathcal{F}'(X_k)$ for all $k \in \mathbb{N}$.

Corollary 7.2.11. *Let* $\hat{X} \in$ *dom* $\mathcal{F}$ *be a solution of* $\mathcal{F}(X) \geq 0$, $\mathcal{F}'(X_k)$ *be stable. Every* X_{k+1}*, determined via a step of the inexact Newton's method (6.1) with* $0 \leq R_k \leq \Phi_{X_k}(X_{k+1})$ *and* $R_k \leq \mathcal{F}(\hat{X})$*, secures the stability of* $\mathcal{F}'(X_{k+1})$*.*

Proof. The concavity of $\mathcal{F}$, as stated in Corollary 7.2.6, implies

$$-\mathcal{F}(X_{k+1}) \leq \mathcal{F}(\hat{X}) - \mathcal{F}(X_{k+1}) \leq \mathcal{F}'(X_{k+1})(\hat{X} - X_{k+1}). \tag{7.14}$$

Corollary 7.2.5 shows

$$\begin{aligned} \mathcal{F}(X_{k+1}) &= \mathcal{F}(X_k) + \mathcal{F}'(X_k)(X_{k+1} - X_k) - \Phi_{X_k}(X_{k+1}) \\ &= R_k - \Phi_{X_k}(X_{k+1}) \leq 0 \end{aligned} \tag{7.15}$$

where the inexact Newton step was exploited. Combining these two results we obtain

$$\begin{aligned} \mathcal{F}'(X_{k+1})(\hat{X} - X_{k+1}) &\geq -\mathcal{F}(X_{k+1}) \\ &= -R_k + \Phi_{X_k}(X_{k+1}) \\ &\geq \Phi_{X_k}(X_{k+1}) \geq 0 \end{aligned} \tag{7.16}$$

as a first result.
Corollary 6.2.2 proves another result, namely $X_{k+1} \geq \hat{X}$.
Now we assume that $\mathcal{F}'(X_{k+1})$ is not stable, which is equivalent to

$$\exists V \in \mathcal{H}_+^n \setminus \{0\}, \beta \geq 0 \quad : \quad \mathcal{F}'(X_{k+1})^*(V) = \beta V \tag{7.17}$$

due to Theorem 3.7 in [19]. Since $X_{k+1} \geq \hat{X}$ we can state

$$\langle V, \mathcal{F}'(X_{k+1})(\hat{X} - X_{k+1})\rangle = \langle \beta V, \hat{X} - X_{k+1}\rangle \leq 0 \tag{7.18}$$

and with (7.16) follows

$$\langle V, \mathcal{F}'(X_{k+1})(\hat{X} - X_{k+1})\rangle \geq \langle V, \Phi_{X_k}(X_{k+1})\rangle \geq 0 \tag{7.19}$$

which results in $\langle V, \Phi_{X_k}(X_{k+1})\rangle = 0$. This can be rewritten in following way

$$\langle V, \mathcal{F}(X_{k+1}) - \mathcal{F}(X_k)\rangle = \langle V, \mathcal{F}'(X_k)(X_{k+1} - X_k)\rangle \tag{7.20}$$

due to Taylor expansion.
Lemma 4.3 in [19] proves $\mathcal{F}'(X_k)^* V = \mathcal{F}'(X_{k+1})^* V = \beta V$ which is a contradiction to the stability of $\mathcal{F}'(X_k)$. Therefore our assumption was wrong and $\mathcal{F}'(X_{k+1})$ is also stable. □

Corollary 7.2.11 introduces a condition on R_k, which secures the inverse negativity in the inexact Newton iterates. Now all requirements of the monotonicity preserving inexact Newton's method can be satisfied, and we can state following convergence result for equation (7.11).

Theorem 7.2.12. *Let $\hat{X} \in D$ satisfy $\mathcal{F}(X) \geq 0$, $\mathcal{F}'(X_0)$ be stable and $R_0 \leq \mathcal{F}(\hat{X})$. Under following requirements on the residuals*

$$\mathcal{F}(X_k) \leq R_k \leq \mathcal{F}(\hat{X}) \qquad k = 1, ... \tag{7.21}$$

$$0 \leq R_k \leq \Phi_{X_k}(X_{k+1}) \qquad k = 0, ... \tag{7.22}$$

$$\lim_{k\to\infty} R_k = 0 \tag{7.23}$$

we obtain monotone iterates $X_1 \geq ... \geq X_k \geq X_{k+1} \geq ... \geq \hat{X}$ for the inexact Newton's method (6.1) which converge to a solution X_∞ of the equation $\mathcal{F}(X) = 0$.

Proof. Condition (7.22) and (7.21) secures the applicability of Corollary 7.2.11, which provides the stability of $\mathcal{F}'(X_k)$ for all inexact Newton iterates $X_k,\ k \in \mathbb{N}$. Corollary 7.2.9 states the equivalence of the stability of $\mathcal{F}'(X_k)$ and the inverse negativity of $\mathcal{F}'(X_k)$. In addition, $\mathcal{F}$ is a concave mapping as shown in Corollary 7.2.6. Therefore, Theorem 6.2.5 can be applied and yields the statements of the Theorem. □

The rational matrix equation, introduced by Damm and Hinrichsen, can be seen as an application of our theory. As a result all special cases, like CARE, DARE and many other, fit to our theory and a monotonicity preserving inexact Newton

method can be established for all this cases.
In addition, we are able to introduce convergence rates in this case. In order to prove a quadratic rate of convergence, we have to claim the stability of $(\mathcal{F}'(X_\infty))^{-1}$.

Theorem 7.2.13. *Let all requirements of Theorem 7.2.12 hold. Additionally the existence of $(\mathcal{F}'(X_\infty))^{-1}$ and the stability of $\mathcal{F}'(X_\infty)$ is assumed. If the residuals satisfy*

$$\|R_k\| \leq \kappa_1 \|X_{k+1} - X_k\|^2 \qquad \forall k \in \mathbb{N},\ \kappa_1 \geq 0. \tag{7.24}$$

then the iterates of method (6.1) will provide a quadratic rate of convergence, i.e. there exists $\kappa \geq 0$ with

$$\|X_{k+1} - X_\infty\| \leq \kappa \|X_k - X_\infty\|^2.$$

Proof. We obtain due to the concavity of $\mathcal{F}$, $\mathcal{F}(X_\infty) = 0$ and (7.15)

$$\begin{aligned} &\mathcal{F}'(X_\infty)(X_\infty - X_{k+1}) \leq \mathcal{F}(X_\infty) - \mathcal{F}(X_{k+1}) = \Phi_{X_k}(X_{k+1}) - R_k \\ \iff &-\mathcal{F}'(X_\infty)(X_{k+1} - X_\infty) \leq \mathcal{F}(X_\infty) - \mathcal{F}(X_{k+1}) = \Phi_{X_k}(X_{k+1}) - R_k \end{aligned}$$

Since $\mathcal{F}'(X_\infty)$ is assumed to be stable, which is equivalent to the inverse positivity of $-\mathcal{F}'(X_\infty)$, see Corollary 7.2.9, we achieve

$$X_{k+1} - X_\infty \leq -\mathcal{F}'(X_\infty)^{-1}(\Phi_{X_k}(X_{k+1}) - R_k).$$

Taking norms we obtain

$$\|X_{k+1} - X_\infty\| \leq \|\mathcal{F}'(X_\infty)^{-1}\| \|\Phi_{X_k}(X_{k+1}) - R_k\|$$

where $\|\cdot\|$ describes the $\mathcal{H}^n$ norm or the induced operator norm. By utilizing the triangle inequality we get

$$\|X_{k+1} - X_\infty\| \leq \|\mathcal{F}'(X_\infty)^{-1}\| (\|\Phi_{X_k}(X_{k+1})\| + \|R_k\|). \tag{7.25}$$

We can use a result of the Damm and Hinrichsen paper, see proof of Theorem 7.3 in [19], and state

$$\|\Phi_{X_k}(X_{k+1})\| \leq \kappa_2 \|X_{k+1} - X_k\|^2 \qquad k \geq 1 \tag{7.26}$$

with $\kappa_2 \geq 0$. Combined with requirement (7.24) and (7.25) we obtain

$$\|X_{k+1} - X_\infty\| \leq \kappa \|X_{k+1} - X_k\|^2 \qquad k \geq 1,\ \kappa \geq 0 \tag{7.27}$$

as a result. Therefore the quadratic rate of convergence is proved. □

Under additional requirements on the residuals and the considered mapping, we were able to state an important convergence result for the rational matrix equation also in case of inexact Newton methods. Of course, this result is also valid for all considered special cases.

7.3 Quasilinearization

The method of quasilinearization has been introduced by Bellman and Kalaba [4] and can be interpreted as Newton's method for a nonlinear differential operator equation [63]. Many generalizations and applications of the quasilinearization technique have been presented in the literature, e.g. in [47].
We try to adapt this idea for the solution of a parabolic partial differential equations (PDE) of the type

$$u_t = u_{xx} + \varphi(u) - f(t,x) \quad \forall (t,x) \in Q_T := (0,T] \times (a,b) \tag{7.28}$$

with initial-

$$u(0,x) = \tilde{u}(x) \qquad \forall x \in \Omega := (a,b) \tag{7.29}$$

and boundary conditions

$$u(t,x) = g(t,x) \quad \forall (t,x) \in \Sigma_T := \{(t,x) |\ t \in (0,T],\ x \in \{a,b\}\}. \tag{7.30}$$

Here $u = u(t,x)$ is a mapping depending on two variables, in which x describes the space variable in an interval $[a,b]$ and t a time variable in $[0,T]$. In addition, the mappings φ, f, $\tilde{u}$ and g are given. A solution of this PDE is a mapping $u : Q_T \to \mathbb{R}^n$ satisfying the dynamics of (7.28) and the conditions (7.29) respectively (7.30).
The quasilinearization approach defines a mapping

$$\mathcal{F}(u) := \begin{pmatrix} u_{xx} - u_t + \varphi(u) - f(t,x) \ \ \forall (t,x) \in Q_T \\ \tilde{u}(x) - u(0,x) \quad \forall x \in \Omega \\ g(t,x) - u(t,x) \ \ \forall (t,x) \in \Sigma_T \end{pmatrix} \tag{7.31}$$

and calculates a solution of $\mathcal{F}(u) = 0$ with help of Newton's method, which is obviously also a solution of the parabolic PDE (7.28) - (7.30). Given an initial iterate $u_0(t,x)$ the k- th Newton step reads as follows

$$\mathcal{F}'(u_k)(u_{k+1} - u_k) = -\mathcal{F}(u_k) \qquad k \in \mathbb{N}.$$

Here the question arise whether above system of equations can be solved within an inexact Newton framework. In order to retain monotone iterates, we have to analyze the requirements of the monotonicity preserving convergence theory, already presented in Chapter 6.3.
At first we analyze the convexity of the mapping $\mathcal{F}$, depending on the structure of φ.

Lemma 7.3.1. *The nonlinear mapping $\mathcal{F}$, defined in (7.31), is convex as long as φ is convex.*

Proof. A comparison of

$$\mathcal{F}'(v)(w-v) := \begin{pmatrix} w_{xx} - v_{xx} - w_t + v_t + \varphi'(v)(w-v) \quad \forall (t,x) \in Q_T \\ -w(0,x) + v(0,x) \qquad \forall x \in \Omega \\ -w(t,x) + v(t,x) \quad \forall (t,x) \in \Sigma_T \end{pmatrix}$$

and

$$\mathcal{F}(w) - \mathcal{F}(v) = \begin{pmatrix} w_{xx} - w_t + \varphi(w) - f(t,x) \quad \forall (t,x) \in Q_T \\ \tilde{u}(x) - w(0,x) \qquad \forall x \in \Omega \\ g(t,x) - w(t,x) \quad \forall (t,x) \in \Sigma_T \end{pmatrix} - \begin{pmatrix} v_{xx} - v_t + \varphi(v) - f(t,x) \quad \forall (t,x) \in Q_T \\ \tilde{u}(x) - v(0,x) \qquad \forall x \in \Omega \\ g(t,x) - v(t,x) \quad \forall (t,x) \in \Sigma_T \end{pmatrix}$$

leads to

$$\begin{gathered} \mathcal{F}'(v)(w-v) \leq \mathcal{F}(w) - \mathcal{F}(v) \\ \Longleftrightarrow \\ \varphi'(v)(w-v) \leq \varphi(w) - \varphi(v). \end{gathered}$$

As a result, the convexity of φ induces the convexity of $\mathcal{F}$. □

In a second step we consider the inverse negativity of $\mathcal{F}'(u_k)$, see Definition 6.1.13. Our goal is to state conditions on $\mathcal{F}$, such that $\mathcal{F}'(u_k)w \geq 0$ implies $w \leq 0$ for all Newton iterates $u_k(t,x),\ k \in \mathbb{N}_0$.
We assume

$$\mathcal{F}'(u_k)w = \begin{pmatrix} w_{xx} - w_t + \varphi'(u_k)w \quad \forall (t,x) \in Q_T \\ -w(0,x) \qquad \forall x \in \Omega \\ -w(t,x) \quad \forall (t,x) \in \Sigma_T \end{pmatrix} = \begin{pmatrix} \tilde{a} \\ \tilde{b} \\ \tilde{c} \end{pmatrix} \geq 0, \tag{7.32}$$

which describes a parabolic PDE for w.
Here we are able to apply the maximum principle. We present a formulation of the maximum principle taken from [39, p.96], applied to (7.32). A more generalized version can be found e.g. in [26].

Theorem 7.3.2. *Assume $\varphi'(u_k) < 0$. If $\tilde{a} \geq 0$ $[\tilde{a} \leq 0]$ then all non-constant solutions w of (7.32) will attain their positive maximum [negative minimum], if existent, on the extended boundary $\Sigma_T \cup \{(0,x) |\ x \in \Omega\}$ of Q_T.*

The maximum principle and the structure of (7.32) enables us to state following Lemma concerning the inverse negativity of $\mathcal{F}'(u_k)$.

Lemma 7.3.3. *Let φ be strictly monotone decreasing. The linear mapping $\mathcal{F}'(u_k)$ is inverse negative for all Newton iterates $u_k,\ k \in \mathbb{N}$.*

Proof. We consider $\mathcal{F}'(u_k)w \geq 0$, as defined in (7.32). If φ is strictly monotone decreasing we will obtain $\varphi'(u_k) < 0$. Now (7.32) defines a parabolic PDE, whose characteristics fit to the maximum principle. Theorem 7.3.2 ensures that the positive maximum, if existent, lies on the boundary $\{(0,x)|\ x \in \Omega\} \cup \Sigma_T$ of Q_T. In addition, the function value on the boundary is known due to (7.32). We obtain $w(t,x) = -\tilde{b}$ for all $(t,x) \in \{(0,x)|\ x \in \Omega\}$ respectively $w(t,x) = -\tilde{c}$ for all $(t,x) \in \Sigma_T$, which are both negative. As a result, there exists no positive value of the solution of the parabolic PDE (7.32) and hence $w \leq 0$ for all solutions of the parabolic PDE (7.32). Therefore $\mathcal{F}'(u_k)w \geq 0$ always implies $w \leq 0$, which is the definition of the inverse negativity of $\mathcal{F}'(u_k)$. □

An additional requirement of the monotonicity preserving convergence theory, presented in chapter 6, is the existence of a solution w of the corresponding inequality $\mathcal{F}(w) \leq 0$.

Lemma 7.3.4. *Let $\tilde{u}(x) \leq 0\ \forall x \in [a,b]$, $g(t,x) \leq 0\ \forall (t,x) \in \Sigma_T$ and $\varphi(0) \leq f(t,x)\ \forall (t,x) \in Q_T$. For w defined as the zero function $w \equiv 0$, we obtain $\mathcal{F}(w) \leq 0$.*

Proof. Due to

$$\mathcal{F}(0) = \begin{pmatrix} \varphi(0) - f(t,x) & \forall (t,x) \in Q_T \\ \tilde{u}(x) & \forall x \in [a,b] \\ g(t,x) & \forall (t,x) \in \Sigma_T \end{pmatrix} \tag{7.33}$$

and the restrictions defined in the Lemma, we achieve $\mathcal{F}(0) \leq 0$. □

The conditions, listed in Lemma 7.3.1, Lemma 7.3.3 and Lemma 7.3.4, guarantee that $\mathcal{F}$, defined in (7.31), satisfies all requirements of our theory. Lemma 7.3.1 secures the convexity of $\mathcal{F}$, Lemma 7.3.3 the inverse negativity of $\mathcal{F}'(u_k)$ for all Newton iterates u_k. Finally Lemma 7.3.4 states a solution of the inequality $\mathcal{F}(w) \leq 0$. Therefore all requirements of Theorem 6.3.3 are met and an inexact version of Newton's method can be applied, which shows a monotone convergence behaviour. In addition, the initial iterate does not have to lie in a neighborhood of a solution.
Unfortunately Newton's method can be rarely realized in infinite dimensional function spaces, apart from toy problems see e.g. [21]. Therefore we concentrate on the discretized problem for (7.31) in a finite dimensional space. Since our considerations indicate that the mapping $\mathcal{F}$ in the infinite dimensional function

space fits to our theory, we expect the discretized version to behave likewise.
We discretize $\bar{Q}_T$ of the parabolic PDE (7.28) - (7.30) with step size $\triangle x = h = \frac{b-a}{n+1}$ in space direction, i.e $x_0 = a$, $x_i = a + ih$ for $i = 1, ..., n$ and $x_{n+1} = b$. The time dimension is discretized with step size $\triangle t = k = \frac{T}{m+1}$ into $t_0 = 0$, $t_i = ik$ for $k = 1, ..., m$ and $t_{m+1} = T$.
With $u_{i,j}$ we now denote the function value $u(t_i, x_j)$, evaluated in one discretization point (t_i, x_j).
The second derivative $u_{xx}(t_i, x_j)$ is approximated with standard finite differences for a fixed time

$$u_{xx}(t_i, x_j) \approx \frac{1}{h^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1})$$

and $u_t(t_i, x_j)$ is estimated by

$$u_t(t_i, x_j) \approx \frac{1}{k}(u_{i+1,j} - u_{i,j}).$$

Now a step of the implicit Euler scheme, see e.g. [71], for the solution of the occuring PDE (7.28) reads as follows

$$\frac{1}{k}(u_{i+1,j} - u_{i,j}) = \frac{1}{h^2}(u_{i+1,j-1} - 2u_{i+1,j} + u_{i+1,j+1}) + \varphi(u_{i+1,j}) - f(t_{i+1}, x_j),$$

which can be rewritten in

$$u_{i+1,j} = u_{i,j} + \frac{k}{h^2}(u_{i+1,j-1} - 2u_{i+1,j} + u_{i+1,j+1}) + k\varphi(u_{i+1,j}) - kf(t_{i+1}, x_j). \tag{7.34}$$

Since the boundary conditions are given in a Dirichlet representation, the function values on the boundary can be easily substituted.
Some abbreviation are necessary for an easier understanding, we therefore define $r := k/h^2$,

$$\vec{u}_0 := \begin{pmatrix} \tilde{u}(x_1) \\ \vdots \\ \tilde{u}(x_n) \end{pmatrix} \in \mathbb{R}^n, \ \vec{u}_j := \begin{pmatrix} u_{j,1} \\ \vdots \\ u_{j,n} \end{pmatrix} \in \mathbb{R}^n, j = 1, ..., m, \ \vec{u} := \begin{pmatrix} \vec{u}_1 \\ \vdots \\ \vec{u}_m \end{pmatrix} \in \mathbb{R}^{mn}$$

and

$$D := \begin{pmatrix} -2 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

In addition, we need $\Phi : \mathbb{R}^n \to \mathbb{R}^n$, defined by

$$\Phi \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} := \begin{pmatrix} \varphi(u_1) \\ \vdots \\ \varphi(u_n) \end{pmatrix} \quad and \quad \vec{f_j} := \begin{pmatrix} f(t_j, x_1) - \frac{1}{h^2} g(t_j, a) \\ f(t_j, x_2) \\ \vdots \\ f(t_j, x_{n-1}) \\ f(t_j, x_n) - \frac{1}{h^2} g(t_j, b) \end{pmatrix}.$$

Utilizing (7.34) and above abbreviations, we state the complete time step for the j-th step of the implicit Euler scheme

$$\vec{u}_{j+1} = \vec{u}_j + rD\vec{u}_{j+1} + k\Phi(\vec{u}_{j+1}) - k\vec{f}_{j+1}. \tag{7.35}$$

This single step can be also presented as solution of a corresponding equation

$$F_j(\vec{u}_{j+1}) = \vec{u}_j - \vec{u}_{j+1} + rD\vec{u}_{j+1} + k\Phi(\vec{u}_{j+1}) - k\vec{f}_{j+1} = 0. \tag{7.36}$$

Our goal is to define a mapping $\mathcal{F}$ such that a solution u_* of $\mathcal{F}(u) = 0$ is also a solution of the discretized problem, i.e. it should cover all time steps of the implicit scheme.
We use the the definition of the j- th time step of the implicit Euler scheme (7.35) and the initial condition to obtain

$$\mathcal{F} \begin{pmatrix} \vec{u}_1 \\ \vdots \\ \vec{u}_j \\ \vdots \\ \vec{u}_m \end{pmatrix} = \begin{pmatrix} \vec{u}_0 - \vec{u}_1 + rD\vec{u}_1 + k\Phi(\vec{u}_1) - k\vec{f}_1 \\ \vdots \\ \vec{u}_{j-1} - \vec{u}_j + rD\vec{u}_j + k\Phi(\vec{u}_j) - k\vec{f}_j \\ \vdots \\ \vec{u}_{m-1} - \vec{u}_m + rD\vec{u}_m + k\Phi(\vec{u}_m) - k\vec{f}_m \end{pmatrix} = 0. \tag{7.37}$$

This mapping $\mathcal{F}(\cdot) : \mathbb{R}^{mn} \to \mathbb{R}^{mn}$ is almost affine-linear, only the parts involving $k\Phi(\vec{u}_j),\ j = 1, ..., m$ are nonlinear. We introduce a partial ordering on $\mathbb{R}^{mn}$ and call $x \leq y$ if and only of $x_i \leq y_i$ for all $i = 1, ..., mn$.
Again we use Newton's method for the solution of $\mathcal{F}(u) = 0$. The Newton iterates $\vec{u}^{\,k},\ k \in \mathbb{N}$ are now elements of $\mathbb{R}^{mn}$ and a Newton step is defined as follows

$$\mathcal{F}'(\vec{u}^{\,k})(\vec{u}^{\,k+1} - \vec{u}^{\,k}) = -\mathcal{F}(\vec{u}^{\,k}) \qquad k \in \mathbb{N},$$

assumed a given initial iterate $\vec{u}^{\,0} \in \mathbb{R}^{mn}$.
Now we focus on the question whether above system of equation fit to our theory, developed in chapter 6. In this case an inexact version of Newton's method would provide a monotone convergence property and a more global convergence.
In a first step we show that $\mathcal{F}'(\vec{u}^{\,k})$ is inverse negative for all Newton iterates $\vec{u}^{\,k},\ k \in \mathbb{N}$.

Lemma 7.3.5. *Assume that φ is strictly monotone decreasing. Then $\mathcal{F}'(\vec{u}^{\,k})$ is inverse negative for all Newton iterates $\vec{u}^{\,k},\ k \in \mathbb{N}$.*

Proof. We consider $\mathcal{F}'(\vec{u}^{\,k})$, defined by

$$\mathcal{F}'(\vec{u}^{\,k}) := -\begin{pmatrix} I - rD - kM_1^k & & & 0 \\ -I & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -I & I - rD - kM_m^k \end{pmatrix} \in \mathbb{R}^{mn \times mn}, \tag{7.38}$$

where

$$M_j^k := \begin{pmatrix} \varphi'(u_{j,1}^k) & & 0 \\ & \ddots & \\ 0 & & \varphi'(u_{j,n}^k) \end{pmatrix} \in \mathbb{R}^{n \times n},\ j = 1, ..., m.$$

Since φ is assumed to be strictly monotone decreasing, we obtain $\varphi'(u_{i,j}^k) < 0$ for every discretization point (t_i, x_j) and each Newton iterate $\vec{u}^{\,k}$. Furthermore, $\mathcal{F}'(\vec{u}^{\,k})$ is of the form $\mathcal{F}'(\vec{u}^{\,k}) = -J_k$, where J_k is a M-matrix, see definition 7.1.4. The M-matrix property can be shown with help of the diagonal dominance of J_k. The diagonal elements J_k are always of the type $1 + 2r - k\varphi(u_{i,i}^k)$ and positive. In the corresponding column, the non-zero elements sum up to $-1 - r - r$ in the worst case. Due to the positivity of $-k\varphi(u_{i,i}^k)$, the absolute value of the diagonal elements is always greater than the sum of the absolute values of the non-diagonal entries in the corresponding column. As a result, we obtain the M- matrix property of J_k with help of Theorem 2.4.14 in [56]. Theorem 7.1.11 states the existence of J_k^{-1} and $J_k^{-1} \geq 0$. Now $\mathcal{F}'(\vec{u}^{\,k})w = -J^k w \geq 0$ always implies $w \leq 0$ due to $J_k^{-1} \geq 0$, which states the inverse negativity of $\mathcal{F}'(\vec{u}^{\,k})$ for all Newton iterates $\vec{u}^{\,k}$. □

Remark 7.3.6. *Lemma 7.3.5 is the analogon of Lemma 7.3.3 in the finite dimensional space.*

The convexity of $\mathcal{F}$ is dependent on the structure of φ.

Lemma 7.3.7. *Assume that φ is convex, then the mapping $\mathcal{F}$ is also convex.*

Proof. Again, we compare

$$\mathcal{F}'(v)(w - v) = -\begin{pmatrix} I - rD - kM_1 & & & 0 \\ -I & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -I & I - rD - kM_m \end{pmatrix} (w - v),$$

where

$$M_j := \begin{pmatrix} \varphi'(v_{j,1}) & & 0 \\ & \ddots & \\ 0 & & \varphi'(v_{j,n}) \end{pmatrix}$$

with

$$\mathcal{F}(w) - \mathcal{F}(v) = \begin{pmatrix} \vec{u}_0 - \vec{w}_1 + rD\vec{w}_1 + k\Phi(\vec{w}_1) - k\vec{f}_1 \\ \vdots \\ \vec{w}_{j-1} - \vec{w}_j + rD\vec{w}_j + k\Phi(\vec{w}_j) - k\vec{f}_j \\ \vdots \\ \vec{w}_{m-1} - \vec{w}_m + rD\vec{w}_m + k\Phi(\vec{w}_m) - k\vec{f}_m \end{pmatrix} - \begin{pmatrix} \vec{u}_0 - \vec{v}_1 + rD\vec{v}_1 + k\Phi(\vec{v}_1) - k\vec{f}_1 \\ \vdots \\ \vec{v}_{j-1} - \vec{v}_j + rD\vec{v}_j + k\Phi(\vec{v}_j) - k\vec{f}_j \\ \vdots \\ \vec{v}_{m-1} - \vec{v}_m + rD\vec{v}_m + k\Phi(\vec{v}_m) - k\vec{f}_m \end{pmatrix}.$$

We obtain

$$\mathcal{F}'(v)(w - v) \leq \mathcal{F}(w) - \mathcal{F}(v)$$

if and only if

$$M_j(\vec{w}_j - \vec{v}_j) \leq \Phi(\vec{w}_j) - \Phi(\vec{v}_j)$$
$$\Longleftrightarrow$$
$$\begin{pmatrix} \varphi'(v_{j,1}) & & 0 \\ & \ddots & \\ 0 & & \varphi'(v_{j,n}) \end{pmatrix} \begin{pmatrix} w_{j,1} - v_{j,1} \\ \vdots \\ w_{j,n} - v_{j,n} \end{pmatrix} \leq \begin{pmatrix} \varphi(w_{j,1}) \\ \vdots \\ \varphi(w_{j,n}) \end{pmatrix} - \begin{pmatrix} \varphi(v_{j,1}) \\ \vdots \\ \varphi(v_{j,n}) \end{pmatrix}.$$

holds for $j = 1, ..., m$. Since φ is assumed to be convex, above relation is always satisfied. Therefore the convexity of φ implies the convexity of $\mathcal{F}$. □

Remark 7.3.8. *Lemma 7.3.7 is the analogon of Lemma 7.3.1 in the finite dimensional space.*

A solution of the inequality $\mathcal{F}(u) \leq 0$ can be found under additional requirements on the structure of $\mathcal{F}$.

Lemma 7.3.9. *Let $\vec{u}_0 + k\Phi(0) - k\vec{f}_1 \leq 0$ and $\Phi(0) - \vec{f}_j \leq 0$ for all $j = 2, ..., m$. Now $u \equiv 0 \in \mathbb{R}^{mn}$ is a solution of $\mathcal{F}(u) \leq 0$.*

Proof. Consider

$$\mathcal{F}(0) = \begin{pmatrix} \vec{u}_0 + k\Phi(0) - k\vec{f}_1 \\ \vdots \\ k\Phi(0) - k\vec{f}_j \\ \vdots \\ k\Phi(0) - k\vec{f}_m \end{pmatrix}, \tag{7.39}$$

together with the assumptions of the Lemma, we obtain $\mathcal{F}(0) \leq 0$. □

Remark 7.3.10. *Lemma 7.3.9 is the equivalence of Lemma 7.3.4 in the finite dimensional space.*

Note, above requirements are less restrictive as those introduced in Lemma 7.3.4.

Lemma 7.3.11. *If the mappings $\varphi,\ f,\ \tilde{u}$ and g satisfy the conditions of Lemma 7.3.4 then the corresponding discretized values will meet all requirements of Lemma 7.3.9.*

Proof. Let $\varphi,\ f,\ \tilde{u}$ and g satisfy the conditions of Lemma 7.3.4. Since $u(0, x) := \tilde{u}(x) \leq 0\ \forall x \in \Omega$, the vector $\vec{u}_0$ of the discretized initial values is negative. In addition, $\varphi(0) \leq f(t, x)\ \forall (t, x) \in Q_T$ and $g(t, x) \leq 0\ \forall (t, x) \in \Sigma_T$, together with the definition and abbreviation of the discretized vectors, we obtain $k\Phi(0) - k\vec{f}_j \leq 0$. □

Now all requirements of the monotonicity preserving convergence theory for inexact Newton methods

$$\mathcal{F}'(\vec{u}^{\,k})(\vec{u}^{\,k+1} - \vec{u}^{\,k}) = -\mathcal{F}(\vec{u}^{\,k}) + R_k \qquad k \in \mathbb{N}, \tag{7.40}$$

are met and we are able to apply Theorem 6.3.3 for the solution of $\mathcal{F}(u) = 0$.

Theorem 7.3.12. *Let $\mathcal{F}(\cdot) : \mathbb{R}^{mn} \to \mathbb{R}^{mn}$ be defined as in (7.37). Assume that φ is convex and strictly monotone decreasing. In addition, assume $\tilde{u}(x) \leq 0\ \forall x \in [a, b],\ g(t, x) \leq 0\ \forall (t, x) \in \Sigma_T$ and $\varphi(0) \leq f(t, x)\ \forall (t, x) \in Q_T$ and $R_0 \leq 0$. Under following requirements on the residuals*

$$-\mathcal{F}(\vec{u}^{\,k}) \leq R_k \leq 0 \qquad k = 1, ... \tag{7.41}$$

$$\lim_{k \to \infty} R_k = 0 \tag{7.42}$$

we obtain monotone iterates $\vec{u}^{\,1} \leq ... \leq \vec{u}^{\,k} \leq \vec{u}^{\,k+1} \leq ... \leq 0$ for the inexact Newton's method (7.40) which converge to a solution of the equation $\mathcal{F}(u) = 0$.

Proof. Since φ is strictly monotone decreasing, we obtain the inverse negativity of $\mathcal{F}'(\vec{u}^{\,k})$ for all inexact Newton iterates $\vec{u}^{\,k}$ with help of Lemma 7.3.5. Furthermore, the convexity of φ implies the convexity of $\mathcal{F}$ due to Lemma 7.3.7. Lemma 7.3.9, together with the requirements on the mappings φ, f, $\tilde{u}$ and g, states $\mathcal{F}(0) \leq 0$. Now we are able to apply Theorem 6.3.3 to conclude the proof. □

As in the infinite dimensional case, we are able to extend Newton's method to an inexact version. Under additional requirements we are able to restore the monotonicity of the iterates and therefore a different convergence radius also for the inexact case. We demonstrated another application of our theory.

Chapter 8

Conclusions and outlook

Our work is initiated with an analysis of algebraic Riccati equations, alike (1.1). Those equations play an important role e.g. in control theory and are therefore of practical interest and well studied. Kleinman [45] already discussed Newton's method for this kind of equation in 1968. His monotonicity and convergence results were unusual and depend on the special structure of the algebraic Riccati equation.

We extended Kleinman's monotonicity and convergence results to alternative versions of inexact Newton methods. Depending on the choice of the applied residuals we achieve a linear, a superlinear or even a quadratic rate of local convergence. A local convergent inexact method can always be developed. Additional requirements on the residuals are necessary to extend the famous convergence results introduced by Kleinman, including monotonicity of the iterates and a global convergence property. Numerical examples demonstrate the benefits of the new developed methods.

Some requirements on the residuals involve matrix inequalities and up to now there is no way to test them efficiently. This is still an open field of research.

A second version of the Kleinman-Newton method has been developed by Banks and Ito [3] and exhibit several advantages compared to the original method, e.g. a low-hand rightside. Unfortunately, we showed that this version is incompatible with inexact Newton methods because it is no longer self-correcting. We therefore recommend the use of the original Kleinman-Newton method in combination with inexact Newton methods.

For linear quadratic regulator problems, one of the major areas of control problems, we are not interested in the solution of an algebraic Riccati equation, only the so-called feedback gain matrix is of importance. Due to modification on the stopping criteria we were able to extend feedback gain algorithms to inexact versions.

Since our results were very promising for algebraic Riccati equation, we analysed similar areas of application. Instead of introducing new theories for every example, we developed a new theoretical background covering all applications as special cases.
We analysed inexact Newton methods with respect to their capability to provide monotone iterates and an alternative convergence radius. To this goal, we introduced several requirements on a function and on the residuals of an inexact Newton method. In order to demonstrate the benefits of the new developed theory we showed the applicability for different applications, e.g. non-symmetric Riccati equation, rational matrix equation occuring in stochastic control and examples taken from the quasilinearization area. As a result, we introduced a new theory and demonstrated its practical importance by several applications.
Finally there should be more applications matching to our theory, which a further analysis will show.

Bibliography

[1] B. D. O. Anderson, *Second-order convergent algorithms for the steady-state Riccati equation*, Intern. J. Control, 28 (1978), pp. 295–306.

[2] B. D. O. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[3] H. Banks and K. Ito, *A numerical algorithm for optimal feedback gains in high dimensional linear quadratic regulator problems*, SIAM Journal on Control and Optimization, 29 (1991), pp. 499–515.

[4] R. Bellman and R. Kalaba, *Quasilinearization and nonlinear boundary-value problems*, Elsevier, New York, 1965.

[5] P. Benner, *Solving large-scale control problems*, IEEE Control Systems Magazine, 24 (2004), pp. 44–59.

[6] P. Benner and R. Byers, *An exact line search method for solving generalized continuous-time algebraic Riccati equations*, IEEE Transactions on Automatic Control, 43 (1998), pp. 101–107.

[7] P. Benner, J. Li, and T. Penzl, *Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems*, Numer. Linear Algebra Appl., 15 (2008), pp. 1–23.

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

[9] ——, *M.E.S.S. 1.0 user manual*, tech. report, Chemnitz Scientific Computing, TU Chemnitz, 2009, to appear [Online] http://www-user.tu-chemnitz.de/ saak/Software.

[10] P. Benner and J. Saak, *Linear-quadratic regulator design for optimal cooling of steel profiles*, tech. report, SFB393/05-05, Sonderforschungsbereich

393 Parallele Numerische Simulation f ür Physik und Kontinuumsmechanik, TU Chemnitz, D-09107 Chemnitz (Germany), 2005[Online] http://www.tu-chemnitz.de/sfb393/sfb05pr.html.

[11] A. Berman and R. J. Plemmons, *Nonnegative matrices in the mathematical sciences*, Academic Press, New York, 1979.

[12] D. A. Bini, B. Iannazzo, and F. Poloni, *A fast newton's method for a nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 30, (2008), pp. 276–290.

[13] A. E. Bouhtouri, D. Hinrichsen, and A. Pritchard, *On the disturbance attenuation problem for a wide class of time invariant linear stochastic systems*, Stochastics Rep. 65, (1991), pp. 255–297.

[14] A. E. Bouhtouri, D. Hinrichsen, and A. Pritchard, *H^∞ type control for discrete-time stochastic systems*, Int. J. Robust Nonlinear Control 9, (1999), pp. 923–948.

[15] C. Broyden, *The convergence of a class of double-rank minimization algorithms*, J. Inst. Math. Appl. 6, (1970), pp. 76–90.

[16] J. Burns, E. Sachs, and L. Zietsman, *Mesh independence of Kleinman-Newton iterations for Riccati equations in Hilbert spaces*, SIAM Journal on Control and Optimization, 47 (2008), pp. 2663–2692.

[17] A. Conn, N. Gould, and P. L. Toint., *Convergence of quasi-newton matrices generated by the symmetric rank one update*, Mathematical Programming 50, (1991), pp. 177–195.

[18] T. Damm, *Rational matrix equations in stochastic control*, Lecture notes in control and information sciences, 297. Springer-Verlag Berlin Heidelberg New York, 2004.

[19] T. Damm and D. Hinrichsen, *Newton's method for a rational matrix equation occuring in stochastic control*, Linear Algebra Appl., 332-334 (2001), pp. 81 – 109.

[20] W. Davidon, *Variable metric method for minimization*, SIAM J. Optim. 1, (1991), pp. 1–17.

[21] P. Deuflhard, *Newton Methods for nonlinear problems: Affine Invariance and Adptive Algorithms*, Springer Series in Computational Mathematics, Vol 35, Springer-Verlag, Berlin Heidelberg New York, 2004.

[22] F. Feitzinger, T. Hylla, and E. Sachs, *Inexact Kleinman-Newton method for Riccati equations*, SIAM Journal on Matrix Analysis and Applications, 31 (2009), pp. 272–288.

[23] R. Fletcher, *A new approach to variable metric algorithms*, Computer Journal 13, (1970), pp. 317–322.

[24] R. Fletcher and M. Powell, *A rapidly convergent descet method for minimization*, Computer Journal 6, (1963), pp. 163–168.

[25] G. Freiling, *A survey of nonsymmetric riccati equations*, Linear Algebra Appl., (2001), pp. 243–270.

[26] A. Friedman, *Partial differential equations of parabolic type*, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[27] Y.-H. Gao and Z.-Z. Bai, *On inexact Newton methods based on doubling iteration scheme for non-symmetric algebraic Riccati equation*, Numer. Linear Algebra Appl., (2010).

[28] D. Goldfarb, *A family of variable metric methods derived by variational means*, Math. Comp. 24, (1970), pp. 23–26.

[29] L. Grasedyck, W. Hackbusch, and B. N. Khoromskij, *Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices*, Computing, 70 (2003), pp. 121–165.

[30] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Num. Analysis 23, (1986), pp. 707–716.

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

[32] C.-H. Guo, *Nonsymmetric algebraic Riccati equations and Wiener-Hopf factorization for M-matrices*, SIAM J. Matrix Anal. Appl., 23, (2001), pp. 225–242.

[33] ——, *On a quadratic matrix equation associated with an M-matrix*, IMA J. Numer. Anal., 23, (2003), pp. 11–27.

[34] C.-H. Guo and N. J. Higham, *Iterative solution of a nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 29, (2007), pp. 396–412.

[35] C.-H. Guo and P. Lancaster, *Analysis and modification of Newton's method for algebraic Riccati equations*, Mathematics of Computation, 67 (1998), pp. 1089–1105.

[36] C.-H. Guo and A. Laub, *On a Newton-like method for solving algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 694–698.

[37] ——, *On the iterative solution of a class of nonsymmetric algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 22, (2000), p. 376391.

[38] D. Guo, Y. J. Cho, and J. Zhu, *Partial ordering methods in nonlinear problems*, Nova Science Publishers, New York, 2004.

[39] G. Hellwig, *Partielle Differentialgleichungen*, Teubner, Stuttgart, 1960.

[40] D. Hinrichsen and A. Pritchard, *Stochastic H^∞*, SIAM J. Control Optim. 36, (1998), pp. 1504–1538.

[41] T. Hylla and E. W. Sachs, *Versions of inexact kleinman-newton methods for riccati equations*, PAMM, Vol. 7, (2008), pp. 1060505–506.

[42] J. Juang, *Existence of algebraic matrix Riccati equations arising in transport theory*, Linear Algebra Appl., 230, (1995), pp. 89–100.

[43] J. Juang and W.-W. Lin, *Nonsymmetric algebraic Riccati equations and hamiltonian-like matrices*, SIAM J. Matrix Anal. Appl., 20, (1998), pp. 228–243.

[44] C. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.

[45] D. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Transactions on Automatic Control, 13 (1968), pp. 114–115.

[46] H. Kwakernaak and R. Sivan, *Linear Optimal control systems*, Wiley, New York, 1972.

[47] V. Lakshmikantham and A. S. Vatsala, *Generalized quasilinearization for nonlinear*, Mathematices Appl., 440 (1998).

[48] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Oxford University Press, New York, 1995.

[49] A. Laub, *A Schur method for solving algebraic Riccati equations*, IEEE Transactions on Automatic Control, 24 (1979), pp. 913– 921.

[50] J. Li, F. Wang, and J. White, *An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect*, in Proc. 36th IEEE/ACM Design Automation Conference, 1999, pp. 1–6.

[51] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.

[52] A. Lu and E. Wachspress, *Solution of Lyapunov equations by alternating direction implicit iteration*, Comp. Math. Appl., 21 (1991), pp. 43–58.

[53] L. A. Lusternik and V. J. Sobolev, *Elements of functional Analysis*, Gordon and Breach, New York, 1961.

[54] V. L. Mehrmann, *The Autonomous Linear Quadratic Control Problem*, Springer, Berlin - Heidelberg, 1991.

[55] K. Morris and C. Navasca, *Approximation of low rank solutions for linear quadratic control of partial differential equations*, Computational Optimization and Applications, (2008).

[56] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[57] D. Peaceman and H. Rachford, *The numerical solution of parabolic and elliptic differential equations*, J. Soc. Indust. and Appl. Math., 3 (1955), pp. 28–41.

[58] T. Penzl, *Lyapack Users Guide*, tech. report, SFB 393/00-33, Sonderforschungsbereich 393 Numerische Simulation auf massiv parallelen Rechnern, TU Chemnitz.

[59] ——, *Numerische Lösung grosser Lyapunov-Gleichungen*, PhD thesis, Fakultät für Mathematik,TU Chemnitz, Berlin, 1998.

[60] ——, *A cyclic low-rank Smith method for large sparse Lyapunov equations*, SIAM Journal on Scientific Computing, 21 (1999), pp. 1401 – 1418.

[61] F. Potra and W. Rheinboldt, *On the monotone convergence of Newton's method*, Computing 36, (1986), pp. 81–90.

[62] J. Roberts, *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, International Journal of Control, 32 (1980), pp. 677–687.

[63] S. Roberts and J. Shipman, *Two-Point Boundary Value Problems: Shooting Methods*, Elsevier, New York, 1972.

[64] I. Rosen and C. Wang, *A multilevel technique for the approximate solution of operator Lyapunov and algebraic Riccati equations*, SIAM Journal on Numerical Analysis, 32 (1995), pp. 514–541.

[65] W. Rudin, *Real and complex analysis, 3rd ed.*, McGraw-Hill, Inc. New York, NY, USA, 1987.

[66] J. Saak, *Effziente numerische Lösung eines Optimalsteuerungsproblems für die Abkühlung von Stahlprofilen*, Diplomarbeit, Fachbereich 3/Mathematik und Informatik Universität Bremen, D-28334 Bremen, (2003).

[67] ——, *Efficient numerical solution of large scale algebraic matrix equations in PDE control and model order reduction*, PhD thesis, TU Chemnitz, 2009.

[68] D. Shanno, *Conditioning of quasi-newton methods for function minimization*, Math. Comp. 24, (1970), pp. 647–656.

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

[70] R. Smith, *Matrix equation $XA + BX = C$*, SIAM Journal on Applied Mathematics, 16 (1968), pp. 198–201.

[71] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 2002.

[72] F. Tröltzsch and A. Unger, *Fast solution of optimal control problems in the selective cooling of steel*, Z. Angew. Math. Mech., (2001), pp. 447–456.

[73] J. Vandergraft, *Newton's method for convex operators in partially ordered spaces*, SIAM J. Numer. Anal., (1967), pp. 406–432.

[74] E. L. Wachspress, *Iterative solution of the Lyapunov matrix equation*, Appl. Math. Letters, 107 (1988), pp. 87–90.

[75] N. Wong and V. Balakrishnan, *Quadratic alternating direction implicit iteration for the fast solution of algebraic Riccati equations*, Proc. Int. Symposium on Intelligent Signal Processing and Communication Systems, (2005), pp. 373– 376.

[76] W. Wonham, *On a matrix Riccati equation of stochastic control*, SIAM J. Control Optim. 6, (1968), pp. 681–698.

Erklärung zur Dissertationsschrift

Hiermit versichere ich, Timo Hylla, dass ich die Dissertationsschrift mit dem Titel

"Extension of inexact Kleinman-Newton methods to a general monotonicity preserving convergence theory"

eigenständig verfasst und keine anderen als die angegebenen Literaturquellen verwendet habe. Die aus fremden Quellen direkt oder indirekt übernommen Gedanken wurden als solche kenntlich gemacht. Diese Dissertationsschrift wurde in gleicher oder ähnlicher Form bisher weder veröffentlicht noch einer anderen Prüfungskommission vorgelegt.

Trier, den 15.11.2010

Dipl. Math. Oec. Timo Hylla