Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Nonlinear multigrid for the solution of large scale Riccati equations in low-rank and $\mathcal{H}$-matrix format

(revised version: May 2007)

by

Lars Grasedyck

Preprint no.: 84 2005

Nonlinear multigrid for the solution of large scale Riccati equations in low-rank and $\mathcal{H}$-matrix format

L. Grasedyck,
Max Planck Institute for Mathematics in the Sciences,
Inselstr. 22-26, 04301 Leipzig, Germany

May 10, 2007

The algebraic matrix Riccati equation $AX + XA^T - XFX + C = 0$, where the matrices $A, B, C, F \in \mathbb{R}^{n\times n}$ are given and a solution $X \in \mathbb{R}^{n\times n}$ is sought, plays a fundamental role in optimal control problems. Large scale systems typically appear if the constraint is described by a partial differential equation. We provide a nonlinear multigrid algorithm that computes the solution X in a data-sparse low rank format and has a complexity of $\mathcal{O}(n)$, subject to the condition that F and C are of low rank and A is the finite element or finite difference discretisation of an elliptic PDE.

We indicate how to generalise the method to $\mathcal{H}$-matrices C, F and X that are only blockwise of low rank and thus allow a broader applicability with a complexity of $\mathcal{O}(n\log(n)^p)$, p a small constant. The method can as well be applied for unstructured and dense matrices C and X in order to solve the Riccati equation in $\mathcal{O}(n^2)$.

Data-sparse approximation, Riccati equation, low rank approximation, multigrid method, hierarchical matrices

1 INTRODUCTION

The (algebraic matrix) Riccati equation

$$A^T X + XA - XFX + C = 0, \qquad A, C, F, X \in \mathbb{R}^{n\times n}, \tag{1}$$

plays an important role in many areas of practical interest, especially optimal control problems [3, 8, 32]. If the constraint of the control is governed by a (linear) partial differential equation, then the discretisation of the PDE will naturally lead to a large scale system. Even for two-dimensional problems a reasonable mesh-width of 1/1000 yields a system matrix of size $n = 10^6$.

The standard discretisation techniques for PDEs lead to *sparse* $n \times n$ system matrices A that contain only $\mathcal{O}(n)$ non-zero entries. This can be exploited for the solution of the linear system $Ax = b$ by iterative methods, e.g., multigrid (cf. [25, 45] and the references therein). The system

is not solved exactly, but with an approximation error in the size of the discretisation error. This allows to compute the approximation in optimal complexity $\mathcal{O}(n)$.

The situation is different for the algebraic matrix Riccati equation where the (typically dense) matrix X contains $\mathcal{O}(n^2)$ entries so that an algorithm for the solution of (1) is regarded optimal if it has a complexity of $\mathcal{O}(n^2)$. In Section 3 we shall prove (based on the results from [2, 15, 38]) that under moderate assumptions on the matrices A, F and C the solution X can be approximated by a matrix $\tilde{X}$ of rank $k = \mathcal{O}(\log(n)\log(1/\varepsilon))$ so that $\|X - \tilde{X}\|_2 \leq \varepsilon \|X\|_2$ holds in the spectral norm $\|\cdot\|_2$. In this scenario an algorithm can be considered optimal if it computes $\tilde{X}$ in $\mathcal{O}(nk)$. The (iterative) algorithm that we propose computes an approximate solution in $\mathcal{O}(nk^2)$ per step, i.e. almost optimal complexity, provided the matrices C and F are of low rank and the matrix A allows for a multilevel treatment (the precise assumptions will be stated later). Numerical examples suggest that the number of steps is $\mathcal{O}(1)$.

The following Section 2 introduces the linear quadratic optimal control problem that serves as our standard example for the structures arising in the Riccati equation. One possible method to solve the nonlinear Riccati equation is to linearise it by Newton's method. This is called the Newton-Kleinman iteration [26]:

$$\begin{array}{lll} X_0 & := & 0 \text{ (or an appropriate stabilising initial guess)} \quad \text{and for } i \in \mathbb{N}_{\geq 1} \\ X_i & \text{solves} & (A - FX_{i-1})^T X_i + X_i(A - FX_{i-1}) + C + X_{i-1}FX_{i-1} = 0. \end{array} \tag{2}$$

In Section 4 we consider the linear multigrid algorithm (and nested iteration) applied to the Lyapunov equations (2) arising in each Newton-Kleinman step. This is an extension of the linear multigrid algorithm that was analysed in [16, 37].

Section 5 contains the nonlinear multigrid algorithm where we theoretically compare the nonlinear Richardson iteration with the linear Richardson iteration in a Newton-Kleinman step. Possible generalisations are collected in Section 6 and the last Section 7 shows the efficiency of our solver applied to the model problem where the matrices are of the size 4190209×4190209. We compare our method with the Newton-Kleinman iteration using linear multigrid to solve the Lyapunov equations in each step, and we compare the linear multigrid with the Cholesky factor ADI algorithm [2, 21, 39, 44].

2 MODEL PROBLEM

The model problem introduced in this Section is the (distributed) control of the two-dimensional heat equation (cf. [28] and the references therein) which is used, e.g., in optimal control problems for the selective cooling of steel [35]. This is a simple academic model problem where one can study many effects of the resulting Riccati equation. In particular, the linear part $L(X) := A^T X + XA$ corresponds to the stiffness matrix of a $2d$-dimensional elliptic partial differential operator [16], i.e., the linear operator L is ill-conditioned in the sense that the condition number grows with increasing size n of the matrix $A \in \mathbb{R}^{n \times n}$. Therefore, standard iterative solvers (e.g., Richardson, Jacobi, Gauss-Seidel and SOR) will need a number of iterations that increases with the condition of L.

We will focus on fast solution techniques for a large scale Riccati equation where the underlying partial differential equation is well understood. The domain where the PDE is posed is the unit square. Using a uniform tensor mesh, it allows for a simple discretisation. Of course, the method

that we propose is in no way limited to regular grids or simple PDEs, but it simplifies both the implementation and presentation.

2.1 Continuous Model

We fix the domain $\Omega := (0,1) \times (0,1) \subset \mathbb{R}^2$ and the boundary $\Gamma := \partial\Omega$. The goal is to minimise the quadratic performance index

$$J(u) := \int_0^\infty \left(y(t)^2 + u(t)^2\right) \mathrm{d}t$$

for the control $u \in L^2(0,\infty)$ and the output $y \in L^2(0,\infty)$ of the corresponding control system

$$\left.\begin{array}{lcll} \partial_t x(t,\xi) & = & \partial^2_{\xi_1} x(t,\xi) + \partial^2_{\xi_2} x(t,\xi) + 2\beta\partial_{\xi_2} x(t,\xi) + \eta(\xi)u(t), & \xi \in \Omega, \quad t > 0, \\ x(t,\xi) & = & 0, & \xi \in \Gamma, \quad t > 0, \\ x(0,\xi) & = & x_0(\xi), & \xi \in \Omega, \\ y(t) & := & \int_\Omega \omega(\xi) x(t,\xi) d\xi, & t > 0. \end{array}\right\} \tag{3}$$

The values of η and ω are

$$\eta(\xi) := \begin{cases} \kappa & \xi_1 < 1/2, \\ 0 & \text{otherwise.} \end{cases} \qquad \omega(\xi) := \begin{cases} 1 & \xi_2 > 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

The two parameters $\beta, \kappa \in \mathbb{R}_{>0}$ can be varied, where β steers the non-symmetry of the system and κ the cost of the control. For $\beta = 0$ we are back at the heat equation. Here we focus on a single-input-single-output system, but a generalisation to multiple inputs and multiple outputs is straight-forward.

We seek the optimal control u^* in linear state feedback form

$$u^*(t,\cdot) = \Pi x(t,\cdot),$$

but since an analytic solution is only for special cases available, we construct a sequence of (semi-) discretisations. For each discretisation level $\ell = 0, 1, \ldots$ an approximation Π_ℓ to the operator Π is computed so that $\Pi_\ell \to \Pi$ [4, 28].

2.2 Semidiscretisation by Finite Differences

The differential equation (3) is discretised by finite differences on a uniform mesh of $(0,1)^2$. On level ℓ the mesh consists of $n = n_\ell = (2^{\ell+1} - 1)^2$ interior grid-points $(x_i)_{i=1}^{n_\ell}$ and a mesh width $h = 2^{-\ell-1}$, cf. Figure 1.

We omit the level index ℓ in the following but bear in mind that all vectors, matrices etc. depend on the level ℓ. By ϕ_i we denote the piecewise bilinear interpolant on the mesh with $\phi_i(x_i) = 1$ and

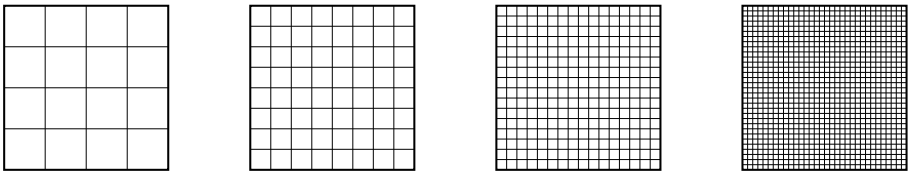

Figure 1: Level $\ell = 1, 2, 3, 4$ of the regular mesh with $n_\ell = 9, 49, 225, 961$ interior nodes.

$\phi_i(x_j) = 0$ for $j \neq i$. The corresponding space-discrete system is

$$\begin{aligned} \partial_t x(t) &= Ax(t) + Ku(t), && t > 0, \\ x(0) &= x_0, && \\ y(t) &:= W^T x(t), && t > 0, \end{aligned} \tag{4}$$

where $A \in \mathbb{R}^{n\times n}$ is the standard finite difference discretisation of the partial differential operator (entries $A_{i,i} = -4h^{-2}$ on the diagonal, $A_{i,j} = h^{-2}$ for neighbouring nodes i, j in the ξ_1 direction and $A_{i,j} = h^{-2} \pm \beta h^{-1}$ for neighbouring nodes i, j in the ξ_2 direction), $x(t) \in \mathbb{R}^n$, $u(t), y(t) \in \mathbb{R}$ and the vectors $K \in \mathbb{R}^n$ and $W \in \mathbb{R}^n$ are

$$K_i := \eta(x_i), \qquad W_j := \int_\Omega \omega(\xi)\phi_j(\xi)\mathrm{d}\xi. \tag{5}$$

Lemma 1 *For $\beta = 0$ the stiffness matrix A is symmetric negative definite. For $\beta > 0$ the matrix A is non-symmetric; the spectrum of A is contained in the left complex halfplane and real-valued as long as $\beta < h^{-1}$.*

Proof: The matrix A has the Kronecker product representation $A = A_1 \otimes I + I \otimes A_2$ where A_1 and A_2 are tridiagonal matrices and $\otimes$ the Kronecker-product (see [16] for more details). Due to [11, 43] the spectrum of A is the sum of the spectra of A_1 and A_2. In [47] the respective spectra of both A_1 and A_2 are explicitly given: that of A_1 corresponding to the one-dimensional discretisation of $\partial^2_{\xi_1}$ is negative real-valued. That of A_2 corresponding to $\partial^2_{\xi_2} + 2\beta\partial_{\xi_2}$ is located in the left complex half-plane, and it is real-valued if $\beta < h^{-1}$. ■

The matrix A from the finite difference discretisation is sparse but ill-conditioned.

In the case that a finite element discretisation is applied we get the space-discrete system

$$E\partial_t x(t) = \tilde{A}x(t) + \tilde{K}u(t), \qquad t > 0, \tag{6}$$

where the entries of the FEM stiffness matrix $\tilde{A}$, mass matrix E and $\tilde{K}$ are

$$\tilde{A}_{ij} = -\int_\Omega \langle \nabla\phi_i, \nabla\phi_j\rangle + \beta\phi_i\partial_2\phi_j, \qquad E_{i,j} = \int_\Omega \phi_i\phi_j, \qquad \tilde{K}_i = \int_\Omega \eta\phi_i.$$

Since the matrix E is well-conditioned (and symmetric) we can write (6) in the form (4) with $A := E^{-1}\tilde{A}$ and $K := E^{-1}\tilde{K}$. Later we will only need to be able to perform the matrix vector

multiplication $z = Ax$ that can be realised by first multiplying $y := \tilde{A}x$ and afterwards solving the system $Ez = y$. Alternatively, one can formulate the following algorithms for the generalised Riccati equation of the form $\tilde{A}^T XE + EX\tilde{A} - XFX + C = 0$. In the following, in order to simplify the presentation, we do not consider the finite element formulation anymore. However, the algorithms that we present work in the same way for finite elements.

2.3 Linear State Feedback Control

In the model problem that we consider, the pair (A, K) is stabilisable and (A, W^T) detectable. (The spectrum of A is already contained in the left complex halfplane so that both conditions are trivially fulfilled). Therefore, the discrete optimal control u can be realised in linear state feedback form [4, 9, 12, 27]

$$u(t) = -K^T X x(t), \qquad t \in [0, \infty),$$

where X is the — in the set of symmetric positive semidefinite matrices — unique solution to the algebraic matrix Riccati equation

$$A^T X + XA - XFX + C = 0, \qquad F := KK^T, \quad C := WW^T. \tag{7}$$

The matrix A is of size $n \times n$. The matrices F and C are of size $n \times n$ and data-sparse in the sense that only K and W have to be stored, i.e., $2n$ entries. The matrix $-K^T X$ is the discrete representation of the feedback operator Π_ℓ on level ℓ.

Remark 1 *The discretisation by finite differences or finite elements will yield a stiffness matrix A that is ill-conditioned, i.e., $\lim_{n\to\infty} \mathrm{cond}_2(A) = \infty$. This has two consequences: first, the Riccati equation is ill-conditioned [7], i.e., small perturbations in the input data may yield large perturbations in the solution. Here, we assume that all the input data is given exact. Second, the Riccati equation will be hard to solve by iterative solvers, and that is why we develop a fast multigrid solver. In Section 7.4 we shall see that the difficulties are even more pronounced when the parameter β governing the non-symmetry of the matrix A is large.*

3 Structure of the Solution

Since the discrete system (4) involves a discretisation error, it is reasonable to solve the Riccati equation only up to an accuracy ε (e.g., of the size of the discretisation error), i.e., we seek an approximation $\tilde{X}$ to the solution X of (1) such that

$$\|X - \tilde{X}\|_2 \le \varepsilon \|X\|_2.$$

The idea now is to choose a matrix $\tilde{X}$ that allows for a data-sparse representation. In the multilevel setting we shall see in Section 4, Remark 2, that the discretisation error can be estimated by the solutions $X_{\ell-1}, X_\ell$ on two subsequent grid levels. A rough estimate suffices because the accuracy ε enters only logarithmically in Theorem 1.

Definition 1 ($R(k)$-matrix representation) *Let $k, n \in \mathbb{N}$. A matrix $M \in \mathbb{R}^{n\times n}$ is called an $R(k)$-matrix (given in $R(k)$-representation) if M is represented in factorised form*

$$M = UV^T, \qquad U, V \in \mathbb{R}^{n\times k}, \tag{8}$$

with U, V in full matrix representation.

The $R(k)$-matrix format is a suitable representation for matrices of rank at most k: each matrix of rank at most k can be written in the factorised form (8) by use of a (reduced) singular value decomposition and each matrix of the form (8) is of rank at most k. The next Theorem proves the existence of a low rank approximant $\tilde{X}$ to the solution X of equation (1), where the rank k is much smaller than the size n of the matrix.

The two factors in the representation (8) of an $R(k)$-matrix involve $2kn$ values to be stored. The matrix-vector multiplication $y := Mx$ can be done in two steps involving the two matrix-vector products $z := V^T x$ and $y := Uz$ that consist of $\mathcal{O}(kn)$ basic arithmetic operations. For $k \ll n$ this is a considerable saving in both memory consumption and computational complexity.

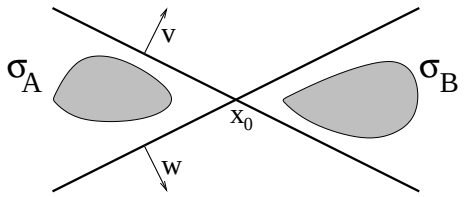

Figure 2: The spectra σ_A and σ_B of A and B are separated.

Theorem 1 (Existence of a low rank approximant) *Let $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ be matrices with spectra contained in two disjoint convex sets σ_A, σ_B of distance $\delta := \operatorname{dist}(\sigma_A, \sigma_B)$. Let $x_0, v, w \in \mathbb{C}$, $|v| = |w| = 1, \langle v, w\rangle \in [0, 1)$, such that (cf. Figure 2)*

$$\begin{aligned} \sigma_A &\subset \{x \in \mathbb{C} \mid \langle x - x_0, v\rangle < 0 \text{ and } \langle x - x_0, w\rangle < 0\}, \\ \sigma_B &\subset \{x \in \mathbb{C} \mid \langle x - x_0, v\rangle > 0 \text{ and } \langle x - x_0, w\rangle > 0\}. \end{aligned}$$

Let Γ_A, Γ_B be paths around the spectrum of A and B with distance at least $\delta/3$ to σ_A, σ_B and between Γ_A, Γ_B,

$$\kappa_A := \frac{1}{2\pi} \oint_{\Gamma_A} \|(\xi I - A)^{-1}\|_2 d\xi, \qquad \kappa_B := \frac{1}{2\pi} \oint_{\Gamma_B} \|(\xi I - B)^{-1}\|_2 d\xi.$$

Let $F \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{n\times m}$ be of rank at most k_F and k_C. Then for each $0 < \varepsilon < 1$ and any matrix $X \in \mathbb{R}^{n\times m}$ that solves

$$AX - XB - XFX + C = 0$$

there exists a matrix $\tilde{X} \in \mathbb{R}^{n\times m}$ that approximates the solution X by

$$\|X - \tilde{X}\|_2 \le \varepsilon \|X\|_2, \tag{9}$$

where the rank of $\tilde{X}$ is bounded by $\operatorname{rank}(\tilde{X}) \le k_\varepsilon k_\sigma (k_C + k_F)$,

$$\begin{aligned} k_\varepsilon &= \mathcal{O}\left(\log\left(1 + \frac{(\|A\|_2 + \|B\|_2)\kappa_A \kappa_B}{\delta\varepsilon}\right)\right), \\ k_\sigma &= \mathcal{O}\left(\left(1 + \tan(\frac{\pi}{2} < v, w >)\right) \log\left(\frac{\operatorname{diam}(\Gamma_A) + \operatorname{diam}(\Gamma_B)}{\delta}\right)\right). \end{aligned}$$

For the model problem (4) from Section 2 ($\beta = 0$) we have $k_\varepsilon k_\sigma = \mathcal{O}(\log^2 n + \log n \log \varepsilon)$.

	level $\ell=3, n_\ell=961$		$\ell=4, n_\ell=3969$		$\ell=5, n_\ell=16129$		$\ell=6, n_\ell=65025$	
k	ε	ε_{K^TX}	ε	ε_{K^TX}	ε	ε_{K^TX}	ε	ε_{K^TX}
1	3×10^{-2}	7×10^{-3}	3×10^{-2}	7×10^{-3}	3×10^{-2}	7×10^{-3}	3×10^{-2}	7×10^{-3}
5	6×10^{-5}	1×10^{-5}	1×10^{-4}	2×10^{-5}	1×10^{-4}	2×10^{-5}	1×10^{-4}	2×10^{-5}
10	5×10^{-9}	1×10^{-9}	7×10^{-8}	1×10^{-8}	2×10^{-7}	3×10^{-8}	4×10^{-7}	5×10^{-8}
15	3×10^{-13}	1×10^{-15}	3×10^{-11}	8×10^{-13}	4×10^{-10}	4×10^{-11}	2×10^{-9}	4×10^{-10}

Table 1: The relative error $\varepsilon := \|X-\tilde{X}\|_2/\|X\|_2$ (and $\varepsilon_{K^TX} := \|K^TX - K^T\tilde{X}\|_2/\|K^TX\|_2$) of a rank k best approximation $\tilde{X}$ of X on different discretisation levels $\ell \in \{3,4,5,6\}$.

Proof: We define the matrix $D := C - XFX$. Then X solves the Sylvester equation $AX - XB + D = 0$. Let $k_D := k_C + k_F$. According to [15, Theorem 1] (see also [38] for the symmetric Lyapunov case and [2] for a generalisation) there exists a matrix $\tilde{X}$ of rank at most

$$\begin{aligned}
\operatorname{rank}(\tilde{X}) &\leq k_\varepsilon k_\sigma k_D, \\
k_\varepsilon &= \left\lceil \log_2\left(1 + \frac{6(\|A\|_2 + \|B\|_2)\kappa_A\kappa_B}{\delta\varepsilon}\right)\right\rceil, \\
k_\sigma &= \mathcal{O}\left(\left(1+\tan(\frac{\pi}{2} < v, w >)\right)\log_2\left(2 + \frac{\operatorname{diam}(\Gamma_A)+\operatorname{diam}(\Gamma_B)}{\delta}\right)\right),
\end{aligned}$$

that approximates X with relative accuracy $\|X-\tilde{X}\|_2 \leq \varepsilon\|X\|_2$. The term k_σ shows the dependency on the location of the spectra of A and B, whereas both k_ε and k_σ are independent of D and X. For the model problem (with $\beta = 0$) the matrix A is symmetric so that [15, Corollary 1] yields

$$\begin{aligned}
k_\varepsilon &= \mathcal{O}\left(\log_2(1/\varepsilon) + \log_2\left(2 + 2\|A\|_2/\delta\right)\right) = \mathcal{O}\left(\log_2(1/\varepsilon) + \log_2\left(n\right)\right), \\
k_\sigma &= \mathcal{O}\left(\log_2\left(2 + \frac{\operatorname{diam}(\Gamma_A)+\operatorname{diam}(\Gamma_B)}{\delta}\right)\right) = \mathcal{O}\left(\log_2\left(n\right)\right).
\end{aligned}$$

■

The dependency of k_ε on $\log(n)$ can be seen in Table 1 where we computed best approximations $\tilde{X}$ of rank $k \in \{1,5,10,15\}$ to the solution X of the Riccati equation (1) for our model problem from Section 2.3 with parameter $\beta = 0$ and $\kappa = 1000$. We observe that for a fixed rank the relative approximation error increases as the level number (and hence the system size n) increases.

The dependency on the location of the spectrum can be seen in Table 2, where we fix the level $\ell = 4$ ($n_\ell = 3969$) and increase the parameter β so that the spectrum of A becomes complex and approaches the imaginary axis. Still, there is an exponential decay in the singular values of the solution X so that it can be approximated by rank $k \ll n$.

Blockwise low rank structures ($\mathcal{H}$-matrices) will be considered in Section 6. In the following section we shall develop an iterative solver (a multigrid method) that computes an approximation $\tilde{X}$ to the solution X of the Riccati equation without ever forming the exact solution. Instead, all iterates are kept in the $R(k)$-matrix format.

k	$\beta = 20$	$\beta = 40$	$\beta = 80$	$\beta = 160$	$\beta = 320$
1	5.7×10^{-2}	8.1×10^{-2}	1.0×10^{-1}	1.1×10^{-1}	1.1×10^{-1}
5	8.1×10^{-4}	1.7×10^{-3}	3.0×10^{-3}	4.4×10^{-3}	6.0×10^{-3}
10	3.0×10^{-6}	1.4×10^{-5}	6.7×10^{-5}	2.3×10^{-4}	5.8×10^{-4}
15	5.5×10^{-9}	4.1×10^{-8}	6.4×10^{-7}	8.0×10^{-6}	5.8×10^{-5}

Table 2: The relative error $\varepsilon := \|X - \tilde{X}\|_2/\|X\|_2$ of a rank k best approximation $\tilde{X}$ of X for different parameters $\beta \in \{20, 40, 80, 160, 320\}$ and fixed $\kappa = 1000$.

4 LINEAR MULTIGRID

The multigrid method to solve an equation of the form

$$Mx = b, \qquad M \in \mathbb{R}^{N\times N},$$

consists of three components (see [25, 45] for an introduction):

1. A hierarchy of discrete problems

   $$M_\ell x_\ell = b_\ell, \quad M_\ell \in \mathbb{R}^{N_\ell \times N_\ell}, \qquad \ell = 1 \ldots, \ell_{\max},$$

   where $N_1 < \cdots < N_{\ell_{\max}} = N$ so that the coarsest problem of size N_1 allows for a direct solution and the finest problem $M_\ell = M$ is the system to be solved.

2. Prolongation and restriction operators

   $$P_{\ell+1\leftarrow\ell} : \mathbb{R}^{N_\ell} \to \mathbb{R}^{N_{\ell+1}}, \qquad R_{\ell-1\leftarrow\ell} : \mathbb{R}^{N_\ell} \to \mathbb{R}^{N_{\ell-1}},$$

   that transfer a vector from one grid level ℓ to the next finer or coarser level $\ell + 1$ or $\ell - 1$.

3. A so-called smoothing iteration

   $$x_\ell^{i+1} = S_\ell(x_\ell^i, b_\ell), \qquad \ell = 1, \ldots, \ell_{\max},$$

   which is not necessarily a good solver but reduces the high-frequency components of the error — hence the name smoother.

The smoothing iteration will be introduced in the next section and the prolongation and restriction operators in Section 4.3. The hierarchy of discretisations of the Riccati equation is given by the hierarchy of finite difference (or finite element) discretisations of the underlying partial differential equation. In this section we denote by the subscript ℓ the level number $\ell = 0, \ldots, \ell_{\max}$. On each level the matrices A_ℓ, K_ℓ, W_ℓ are given and they define the matrices F_ℓ, C_ℓ in the Riccati equation. As a result we get the solution $X_\ell \in \mathbb{R}^{n_\ell \times n_\ell}$ on each level ℓ. The vector representation of the matrix X_ℓ is $x_\ell \in \mathbb{R}^{N_\ell}$, $N_\ell := n_\ell^2$, and the large linear system to be solved in each Newton-Kleinman step (2) is of the form $M = (A - FX_i)^T \otimes I + I \otimes (A - FX_i)^T$ with right-hand side b_ℓ being the vector representation of the matrix $-C - X_i F X_i$.

The idea of the multigrid method is that on each level ℓ the smoother S_ℓ reduces the high-frequency components (relative to the level ℓ) of the error. If this is done on each level, then all components

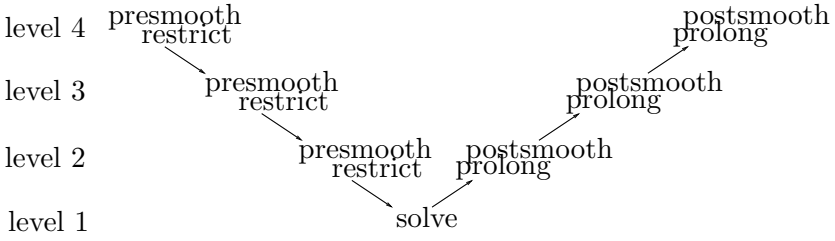

Figure 3: One step of the V-cycle multigrid on $\ell = 4$ levels.

are reduced. One step of the multigrid algorithm (using ν_1 presmoothing steps, ν_2 postsmoothing steps and γ coarse-grid corrections) is given in Algorithm 1. The case $\gamma = 2$ is only relevant for theoretical purposes, in practice we will always use $\gamma = 1$ (the so-called V-cycle multigrid), which is schematically depicted in Figure 3 for $\ell = 4$ levels.

Algorithm 1 Linear Multigrid Step

Procedure multigrid_step(M_ℓ, b_ℓ, ν_1, ν_2, γ, **var** x_ℓ)
if $\ell = 1$ **then**
  Solve $M_\ell x_\ell = b_\ell$ {coarsest grid solve}
else
  for $j = 1, \ldots, \nu_1$ **do**
    $x_\ell := S_\ell(x_\ell, b_\ell)$ {ν_1 presmoothing steps}
  end for
  $b_{\ell-1} := R_{\ell-1 \leftarrow \ell}(M_\ell x_\ell - b_\ell)$ {restriction of the defect to the coarse grid}
  $x_{\ell-1} := 0$
  for $j = 1, \ldots, \gamma$ **do**
    Call **multigrid_step**($M_{\ell-1}$, $b_{\ell-1}$, ν_1, ν_2, $x_{\ell-1}$) {recursive call}
  end for
  $x_\ell := x_\ell - P_{\ell \leftarrow \ell-1}(x_{\ell-1})$ {coarse grid correction}
  for $j = 1, \ldots, \nu_2$ **do**
    $x_\ell := S_\ell(x_\ell, b_\ell)$ {ν_2 postsmoothing steps}
  end for
end if

4.1 Nested Iteration

The so-called nested iteration is used to provide good starting iterates on each level: after computing an approximate solution $x^i_{\ell-1}$ on level $\ell-1$ (e.g., i steps of the multigrid algorithm), one can prolong it to level ℓ

$$x^1_\ell := P_{\ell \leftarrow \ell-1}(x^i_{\ell-1})$$

and then apply again i multigrid steps to obtain an approximate solution x^i_ℓ. Starting from level $\ell = 1$ and going up to $\ell = \ell_{\max}$ we thus need to perform only a minimal number of multigrid steps on the finest level $\ell_{\max}$.

Remark 2 *On each level ℓ the discrete solution $x_\ell \in \mathbb{R}^{N_\ell}$ is the coefficient vector of a function $v(\xi) = \sum_{i=1}^{N_\ell}(x_\ell)_i \varphi_i(\xi), \xi \in \Omega$. We denote the space spanned by these functions by V_ℓ. Then the*

relative discretisation error is defined as

$$\delta_\ell := \min_{w \in V_\ell} \|v - w\|/\|v\|,$$

where $v = \lim_{\ell \to \infty} v_\ell \in V = \cup_{\ell \in \mathbb{N}} V_\ell$ *is the continuous solution and* $\|\cdot\|$ *a suitable norm on* V*. If the (approximate) solutions* $x_{\ell-1}, x_\ell$ *on two subsequent grid levels have been computed, then the relative discretisation error* $\delta_{\ell-1}$ *on level* $\ell - 1$ *can be estimated by*

$$\delta_{\ell-1} = \min_{w \in V_{\ell-1}} \|v - w\|/\|v\| \approx \|v - v_{\ell-1}\|/\|v\| \approx \|v_\ell - v_{\ell-1}\|/\|v_\ell\|.$$

The nested iteration alone (without multigrid) based only on a simple iterative solver (e.g. the smoothing iteration) on each level will not yield a fast algorithm, since the convergence rate of the simple iterative solver on the fine grid will quickly tend to 1, whereas the multigrid iteration has a convergence rate $\rho < 1$ independent of the level ℓ [24, 25, 45].

4.2 Linear Richardson Iteration

A simple iterative solver for large sparse systems $Mx = b$ is the Richardson iteration

$$x^{i+1} := S(x^i, b) := x^i + \lambda(Mx^i - b), \qquad i = 1, \ldots \tag{10}$$

that converges to the solution x for all negative definite matrices M and a damping factor $0 < \lambda < 2/\|M\|_2$, cf. [24]. The optimal damping factor λ_{opt} (which yields the best convergence rate ρ_{opt}) is

$$\lambda_{\text{opt}} = 2/(\|M\|_2 + \|M^{-1}\|_2^{-1}), \qquad \rho_{\text{opt}} = \frac{\|M\|_2 - \|M^{-1}\|_2^{-1}}{\|M\|_2 + \|M^{-1}\|_2^{-1}}. \tag{11}$$

Typically, an estimate for $\|M\|_2$ is known, e.g. by few steps of the power iteration.

The quite simple Richardson iteration is of special interest because it can easily be performed for structured matrices, e.g. low rank matrices in the R(k)-matrix format. Other iterative schemes like Gauss-Seidel, SOR or ILU [24] cannot be easily adapted to rank structured matrices as required for the solution of large scale matrix equations.

The linear Richardson iteration can be applied to solve the linear matrix equations

$$\mathcal{L}_Y(X) := (A^T - YF)X + X(A - FY) = -C - YFY,$$

arising in each step of the Newton-Kleinman iteration. However, the linear operator $X \mapsto \mathcal{L}_Y(X)$ is not necessarily symmetric.

Lemma 2 ([34]) *a) The Richardson iteration for a linear operator* $\mathcal{L}$ *converges for a suitable choice of the damping parameter* λ *(sufficiently small) if the spectrum of* $\mathcal{L}$ *is contained in the left complex halfplane.*
b) For $-\frac{1}{2}(\mathcal{L} + \mathcal{L}^H) \geq c_1 I$ *and* $\mathcal{L}^H\mathcal{L} \leq -c_2\frac{1}{2}(\mathcal{L} + \mathcal{L}^H)$*,* $c_1, c_2 \in \mathbb{R}_{>0}$*, the Richardson iteration converges monotonically in the Euclidean norm with rate* $\rho \leq \sqrt{1 - c_1/c_2}$ *for the damping parameter* $\lambda := 1/c_2$*.*

In practice one can use the following strategy: if $\mathcal{L}$ were symmetric negative definite, then (11) would yield the optimal damping factor $\bar{\lambda}$, and this is an upper bound for the damping factor λ that one may use for $\mathcal{L}$. If the Richardson iteration diverges, then one has to reduce (e.g., halve) the damping factor and restart. In this way, the iteration may become rather slow (due to the fact that the spectrum requires a small λ) but it will eventually converge.

4.3 Prolongation and Restriction

A vector $v_\ell \in \mathbb{R}^{n_\ell}$ on level ℓ of the grid is the discrete representation of a grid-function

$$\mathfrak{v}_\ell(x) = \sum_{i=1}^{n_\ell} (v_\ell)_i \, \varphi_{i,\ell}(x), \qquad x \in \Omega.$$

The so-called *prolongation* $v_{\ell+1}$ of v_ℓ to the next finer level $\ell+1$ is given by evaluating the corresponding grid-function at the grid points $x_{i,\ell+1}$,

$$(v_{\ell+1})_i := \mathfrak{v}_\ell(x_{i,\ell+1}), \quad i \in \{1, \ldots, n_{\ell+1}\},$$

i.e., the grid-data are interpolated as depicted in Figure 4.

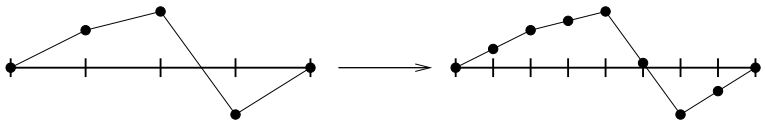

Figure 4: The function on the coarse grid (left) is interpolated linearly in the fine grid points.

The prolongation operator $p_{\ell+1\leftarrow\ell}$ is defined by

$$p_{\ell+1\leftarrow\ell} : \mathbb{R}^{n_\ell} \to \mathbb{R}^{n_{\ell+1}}, \quad v_\ell \mapsto v_{\ell+1}, \quad (v_{\ell+1})_i := \mathfrak{v}_\ell(x_{i,\ell+1}). \tag{12}$$

The adjoint of the prolongation in the scalar product $\langle x, y\rangle_\ell := h_\ell^d \sum_i x_i \bar{y}_i$ is the *restriction* operator $r_{\ell-1\leftarrow\ell}$. In our model problem ($d = 2$) this is $r_{\ell-1\leftarrow\ell} = \frac{1}{4} p^T_{\ell\leftarrow\ell-1} : \mathbb{R}^{n_\ell} \to \mathbb{R}^{n_{\ell-1}}$, which is an averaging of values at fine-grid nodes surrounding the coarse-grid node.

The two grid transfer operators r and p allow for a multilevel representation of vectors defined on the discrete grid. These *vector* transfer operators can be generalised to *matrix* transfer operators R and P.

Definition 2 (Matrix Prolongation and Restriction) *The matrix prolongation operator* $P_{\ell+1\leftarrow\ell} : \mathbb{R}^{n_\ell \times n_\ell} \to \mathbb{R}^{n_{\ell+1} \times n_{\ell+1}}$ *and matrix restriction operator* $R_{\ell-1\leftarrow\ell} : \mathbb{R}^{n_\ell \times n_\ell} \to \mathbb{R}^{n_{\ell-1} \times n_{\ell-1}}$ *are defined by*

$$P_{\ell+1\leftarrow\ell}(X) := p_{\ell+1\leftarrow\ell} \, X \, p^T_{\ell+1\leftarrow\ell}, \quad R_{\ell-1\leftarrow\ell}(Y) := r_{\ell-1\leftarrow\ell} \, Y \, r^T_{\ell-1\leftarrow\ell}.$$

For an $R(k)$-matrix $M = \sum_{i=1}^k u_i v_i^T$ on level ℓ, the prolongation and restriction are immediately given in the $R(k)$-matrix format :

$$P_{\ell+1\leftarrow\ell}(M) = \sum_{i=1}^{k} p_{\ell+1\leftarrow\ell} u_i \, (p_{\ell+1\leftarrow\ell} v_i)^T, \quad R_{\ell-1\leftarrow\ell}(M) = \sum_{i=1}^{k} r_{\ell-1\leftarrow\ell} u_i \, (r_{\ell-1\leftarrow\ell} v_i)^T. \tag{13}$$

Lemma 3 *The complexity of the matrix prolongation and restriction of an* $n \times n$ $R(k)$*-matrix is* $\mathcal{O}(nk)$*, i.e., optimal complexity. For a general full matrix the complexity is* $\mathcal{O}(n^2)$*.*

Proof: The prolongation or restriction operators have to be applied to k vectors each, see (13), where the vectors are of length n. The vector prolongation and restriction operators $p_{\ell+1\leftarrow\ell}$ and $r_{\ell-1\leftarrow\ell}$ are of complexity $\mathcal{O}(n)$ since for each entry i ($i = 1, \ldots, n$) they involve only $\mathcal{O}(1)$ neighbouring nodes in the grid. ■

4.4 Convergence of Linear Multigrid Applied to Newton-Kleinman

In each Newton-Kleinman step for the solution of the Riccati equation

$$\mathcal{R}(X) := A^T X + XA - XFX = -C \tag{14}$$

one has to solve a linear Lyapunov equation

$$\mathcal{L}_Y(X) := (A^T - YF)X + X(A - FY) = -C - YFY, \tag{15}$$

where the spectrum of $A - FY$ (and thus $\mathcal{L}_Y$ [43]) is contained in the left complex halfplane. The matrix Y is the approximation of the Riccati solution X from the previous Newton step. For the solution of this linear system we will apply the linear multigrid method.

Two conditions for the convergence of the linear multigrid method are sufficient: First, the approximation property and second the smoothing property. The approximation property

$$\|M_\ell^{-1} - P_{\ell \leftarrow \ell-1} M_{\ell-1}^{-1} R_{\ell-1 \leftarrow \ell}\| \le C_{\text{app}} / \|M_\ell\|$$

relates two consecutive levels of the discretisation. The operator $\mathcal{L}_0$ ($\mathcal{L}_Y$ for $Y = 0$) is the finite difference (finite element, resp.) discretisation of an elliptic partial differential operator [16]. Under sufficient regularity assumptions (e.g., smooth coefficients and a convex domain like in our model problem) the approximation property holds for the operator $\mathcal{L}_0$ [24].

Remark 3 *The system matrix of the operator $\mathcal{L}_Y$ is*

$$M = (A - FY)^T \otimes I + I \otimes (A - FY)^T = \left(A^T \otimes I + I \otimes A^T\right) - \left((FY)^T \otimes I + I \otimes (FY)^T\right).$$

The first term $A^T \otimes I + I \otimes A^T$ corresponds to the elliptic part $\mathcal{L}_0$ of $\mathcal{L}$. We assume that this part is dominating, since the second term $(FY)^T \otimes I + I \otimes (FY)^T$ is a lower order term. Although we do not have a proof, it is plausible that the approximation property also holds for $\mathcal{L}_Y$.

The smoothing property

$$\|M_\ell \hat{S}_\ell^\nu\|_2 \le \eta(\nu) \|M_\ell\|_2, \quad \lim_{\nu \to \infty} \eta(\nu) = 0,$$

(with $\hat{S}_\ell$ being the iteration matrix of S_ℓ and $\eta(\nu)$ being independent of the level number ℓ) says that first, the smoothing iteration converges, and second, the rate measured in the M_ℓ-norm is level-independent.

Corollary 1 *Let $\mathcal{L}_Y$ denote the Lyapunov operator in the Newton-Kleinman iteration.*
a) The Richardson iteration converges for sufficiently small damping parameter λ.
b) The Richardson iteration with λ as in a) fulfils the smoothing property, i.e. it is a smoother suitable for the multigrid algorithm.
In both cases the hidden constants depend on the location of the spectrum of $\mathcal{L}$, e.g., the convergence rate ρ tends to 1 as $\beta \to \infty$ in our model problem (cf. Section 7.4).

Proof: Part a) follows from Lemma 2 and the fact that in each Newton-Kleinman step the matrix $A - FY$ is stable. Part b) follows from the Lemma of Reusken [40]. ∎

4.5 Complexity of Linear Multigrid Applied to Newton-Kleinman

We assume that the smoothing iteration S_ℓ is of linear complexity $\mathcal{O}(N_\ell)$. This is fulfilled for the Richardson iteration provided that the matrix M can be evaluated in $\mathcal{O}(N_\ell)$. Further we assume that two consecutive grid levels fulfil $N_{\ell-1} \leq N_\ell / C_{red}, C_{red} > 1$. In our model problem this holds for the constant $C_{red} := 4$.

Theorem 2 *One step of the multigrid iteration (Algorithm 1) applied to $\mathcal{L}_Y$ on level ℓ is of linear complexity $\mathcal{O}(N_\ell)$.*

Proof: The smoothing iteration is of linear complexity, i.e., $\mathcal{O}(N_\ell)$ operations are necessary to perform one step $x_\ell \mapsto S_\ell(x_\ell, b_\ell)$. According to Lemma 3 in Section 4.3 the prolongation and restriction operator can be applied using $\mathcal{O}(N_\ell)$ operations. The system matrix $M = (A - FX)^T \otimes I + I \otimes (A - FX)^T$ allows for a matrix by vector multiplication in $\mathcal{O}(N_\ell)$ due to the fact that F is of rank $\mathcal{O}(1)$ and A is sparse (in the FEM case the stiffness matrix $\tilde{A}$ is sparse and the mass matrix E can be inverted in $\mathcal{O}(1)$ iterative steps). According to [24, Theorem 10.4.2] the multigrid method (with $\gamma < C_{red}$ coarse-grid corrections) is of linear complexity. ■

The linear complexity of the multigrid algorithm to solve $Mx = b$ is optimal when applied for general right-hand sides b and solutions x. Since we already know that the right-hand side $-C - YFY$ in each Newton-Kleinman step (15) is of rank at most $\operatorname{rank}(C) + \operatorname{rank}(F)$ ($= 2$ in our model problem), and the solution is according to Theorem 1 of low rank k_X, we will modify the multigrid algorithm in such a way that it has complexity $\mathcal{O}(n_\ell)$ instead of $\mathcal{O}(N_\ell) = \mathcal{O}(n_\ell^2)$.

4.6 Low Rank Linear Multigrid

We exploit the low-rank structure of the approximate solution $\tilde{X}$ from Theorem 1 by approximating all iterates X^i in the low-rank format from Definition 1.

Lemma 4 *Let $\tilde{X}^i$ be an $n \times n$-$R(k)$-matrix, A sparse and F of rank k_F and C of rank k_C. Then one step of the linear Richardson iteration*

$$X^{i+1} := \tilde{X}^i - \lambda((A^T - YF)\tilde{X}^i + \tilde{X}^i(A - FY) + C + YFY)$$

applied to the linear operator $\mathcal{L}_Y$ appearing in the Newton-Kleinman iteration yields a matrix $X^{i+1} \in R(2k + k_C + k_F)$ that can be computed in $\mathcal{O}(nk)$.

Proof: The rank of $\tilde{X}^i - \lambda(A^T - YF)\tilde{X}^i = (I - \lambda(A^T - YF))\tilde{X}^i$ is bounded by k, as well as that of $-\lambda\tilde{X}^i(A - FY)$. This proves the rank bound for X^{i+1}. Let $\tilde{X}^i = \sum_{j=1}^k u_j v_j^T$. For the computation of X^{i+1} we have to compute the $2k$ matrix by vector products $(I - \lambda(A^T - YF))u_j$ and $-\lambda\tilde{X}^i(A^T - YF)v_j$ which can each be performed in $\mathcal{O}(n)$ because A is sparse and F of rank k_F. ■

The exact next iterate X^{i+1} after one Richardson step bears a rank that is increased by a factor of at least 2 so that the rank will rapidly grow if we perform the Richardson iteration in exact arithmetic.

Since we assume that X^i tends to the solution X that can be approximated in the $R(k)$-matrix format with low rank k, we want to approximate the next iterate X^{i+1} by an $R(k)$-matrix $\tilde{X}^{i+1}$. This requires the computation of a projection of X^{i+1} to the set $R(k)$.

Lemma 5 *Let $R = UV^T$ be an $n \times n$ $R(k)$-matrix. Let $U = Q_U R_U$ and $V = Q_V R_V$ be respective QR decompositions with $Q_U, Q_V \in \mathbb{R}^{n\times k}$ and $R_U, R_V \in \mathbb{R}^{k\times k}$. Let*

$$R_U V_U^T = \tilde{U}\Sigma\tilde{V}^T, \qquad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_k),$$

be a singular value decomposition of $R_U V_U^T$. Then for any $k' \in \{1, \ldots, k\}$ the matrix

$$\tilde{R} := \left(Q_U\tilde{U}\tilde{\Sigma}\right)\left(Q_V\tilde{V}\right)^T, \qquad \tilde{\Sigma} := \operatorname{diag}(\sigma_1, \ldots, \sigma_{k'}, 0, \ldots, 0)$$

is a rank k' best approximation of R. It can be computed in $\mathcal{O}(nk^2)$.

Proof: A (reduced) singular value decomposition of R is given by

$$R = UV^T = Q_U R_U R_V^T Q_V^T = \underbrace{Q_U\tilde{U}}_{\text{orthonormal}} \quad \Sigma \quad \underbrace{\tilde{V}^T Q_V^T}_{\text{orthonormal}}$$

so that replacing Σ by $\tilde{\Sigma}$ gives a rank k' best approximation in the Euclidean and Frobenius norm [14]. The two QR decompositions can be computed in $\mathcal{O}(nk^2)$ and the SVD in $\mathcal{O}(k^3)$ [14, 5.2.9 and 5.4.5]. The multiplications are again of complexity $\mathcal{O}(nk^2)$. ∎

The previous Lemma gives rise to the definition of a truncation operator $\mathcal{T}^{k'\leftarrow k}$, cf. [17].

Definition 3 (Truncation operator $\mathcal{T}^{k'\leftarrow k}$) *For $R(k)$-matrices R we define the truncation operator*

$$\mathcal{T}^{k'\leftarrow k}(R) := \tilde{R},$$

where $\tilde{R}$ is a best approximation of R in the set of $R(k')$-matrices. If $\tilde{R}$ is not unique, then $\mathcal{T}^{k'\leftarrow k}(R)$ is an arbitrary representative.

Using the previously defined truncation operator we can formulate the linear low-rank Richardson iteration by

$$\begin{aligned}\tilde{X}^{i+1} &:= S^k(\tilde{X}^i, -C - YFY) \\ &:= \mathcal{T}^{k\leftarrow 2k+k_C+k_F}\left(\tilde{X}^i - \lambda((A^T - YF)\tilde{X}^i + \tilde{X}^i(A - FY) + C + YFY)\right). \qquad (16)\end{aligned}$$

The influence of the truncation on the convergence will be studied in the last Section 7. In principle one can choose a large enough rank k so that the truncation error is of similar size as the machine precision — but this will increase the computational effort.

Since the exact Richardson step is of complexity $\mathcal{O}(nk)$ and the truncation of complexity $\mathcal{O}(nk^2)$, we conclude that one step of the low-rank Richardson iteration is of complexity $\mathcal{O}(nk^2)$. In the multigrid Algorithm 1 we can now use the low-rank Richardson as a smoother with prescribed rank

k. For the defect $b_{\ell-1}$ on the next coarser grid (used in the coarse-grid correction step) we prescribe a rank of $k_{cg} := 2k + k_C + k_F$, i.e.

$$b_{\ell-1} := \mathcal{T}^{k_{cg} \leftarrow k+k_{cg}}(M_\ell x_\ell - b_\ell),$$

and we truncate the iterate after the coarse-grid correction, i.e.

$$x_\ell := \mathcal{T}^{k \leftarrow 2k}(x_\ell - P_{\ell \leftarrow \ell-1}(x_{\ell-1})).$$

Lemma 6 *The complexity of one multigrid step on level ℓ is $\mathcal{O}(n_\ell k^2)$.*

So far we have used the multigrid method in order to solve the linear system $\mathcal{L}_Y$ arising in each step of the Newton-Kleinman iteration, cf. Algorithm 2.

Algorithm 2 Newton-Kleinman Iteration

Procedure nk_iteration($(A_\ell)_{\ell=1}^{\ell_{\max}}$, $(F_\ell)_{\ell=1}^{\ell_{\max}}$, $(C_\ell)_{\ell=1}^{\ell_{\max}}$, m, **var** $X_{\ell_{\max}}$)
{$X_{\ell_{\max}}$ contains the initial guess and m is the number of Newton steps}
for $i = 1, \ldots, m$ **do**
  $Y_{\ell_{\max}} := X_{\ell_{\max}}$
  for $\ell = \ell_{\max}, \ldots, 2$ **do**
    $Y_{\ell-1} := R_{\ell-1 \leftarrow \ell}(Y_\ell)$ {restrict current iterate to coarse levels}
  end for
  $X_1 := Y_1$
  for $\ell = 1, \ldots, \ell_{\max} - 1$ **do**
    Solve $\mathcal{L}_{Y_\ell}(X_\ell) = -C_\ell - Y_\ell F_\ell Y_\ell$ { initial guess X_ℓ }
    $X_{\ell+1} := P_{\ell+1 \leftarrow \ell}(X_\ell)$ { prolong solution to next level }
  end for
  Solve $\mathcal{L}_{Y_{\ell_{\max}}}(X_{\ell_{\max}}) = -C_{\ell_{\max}} - Y_{\ell_{\max}} F_{\ell_{\max}} Y_{\ell_{\max}}$ { initial guess $X_{\ell_{\max}}$ }
end for

The total complexity to solve the Riccati equation in low-rank format is thus $\mathcal{O}(\#\text{Newton steps} \times \#\text{multigrid steps} \times nk^2)$. In the following Section we present two alternatives that solve the Riccati equation in $\mathcal{O}(\#\text{multigrid steps} \times nk^2)$.

5 NONLINEAR MULTIGRID

There are two types of nonlinear multigrid methods to solve the Riccati equation: first, the Newton-multigrid and second the full nonlinear multigrid.

5.1 Newton-Multigrid

The idea of Newton-multigrid is to use a coarse-grid solution as an initial guess of the Newton iteration on each level. This is similar to the nested iteration but additionally m Newton steps are applied on each level. On the finest level $\ell_{\max}$ we have to apply only a few Newton steps in order to reduce the error to the size of the discretisation error. For sufficiently fine grids typically $m = 2$ Newton steps are enough. In order to be able to apply the multigrid method for the linear systems

Algorithm 3 Newton-Multigrid

Procedure nmg_iteration($(A_\ell)_{\ell=1}^{\ell_{\max}}$, $(F_\ell)_{\ell=1}^{\ell_{\max}}$, $(C_\ell)_{\ell=1}^{\ell_{\max}}$, m, X_1 **var** $X_{\ell_{\max}}$)
{ X_1 is the solution on the coarsest grid, m is the number of Newton steps per level }
for $\ell = 2, \ldots, \ell_{\max}$ **do**
  $X_\ell := P_{\ell \leftarrow \ell-1}(X_{\ell-1})$ { prolongation of the solution from the coarse grid }
  for $i = 1, \ldots, m$ **do**
    $Y_\ell := X_\ell$
    Solve $\mathcal{L}_{Y_\ell}(X_\ell) = -C_\ell - Y_\ell F_\ell Y_\ell$ { initial guess X_ℓ }
  end for
end for

arising on each level ℓ, we have to assume that the approximate solution X_ℓ stabilises the system on the next level:

$$A_{\ell+1} - F_{\ell+1} P_{\ell+1 \leftarrow \ell}(X_\ell) \qquad \text{is a stable matrix.}$$

5.2 Full Nonlinear Multigrid

The idea of the full nonlinear multigrid is to apply a smoothing iteration that is able to solve (at least locally) the nonlinear problem. Of course, the convergence might be very slow, but as long as the iteration has the smoothing property (reduces the high-frequency components of the error) one can obtain the fast multigrid convergence. The smoothing iteration that we use is the nonlinear Richardson iteration that will be discussed in Section 5.3. The prolongation and restriction operators used for the nonlinear multigrid method are the same as in the linear case. However, the coarse-grid correction has to be modified as described in Section 5.4.

5.3 Nonlinear Richardson

For nonlinear systems of equations $\mathcal{R}(x) = b$ the nonlinear Richardson iteration

$$x^{i+1} := \mathfrak{S}(x^i, b) := x^i + \lambda_i(\mathcal{R}(x^i) - b), \qquad i = 1, \ldots \tag{17}$$

can be performed, but the damping parameter λ_i has to be determined depending on the iterate x^i by estimating the optimal damping parameter of the linearised operator: for the Riccati operator

$$\mathcal{R}(X^i) := A^T X^i + X^i A - X^i F X^i \tag{18}$$

we consider (for the theory) the corresponding linearisation

$$D\mathcal{R}(X^i)(Y) = (A^T - X^i F)Y + Y(A - F X^i).$$

The linearisation approximates the nonlinear Riccati operator well, at least locally. In the next theorem we shall prove that the nonlinear Richardson iteration behaves (locally) like the linear Richardson iteration applied to the linearised problem. The choice of a suitable damping parameter λ_i for the linear operator $D\mathcal{R}(X^i)$ is discussed in Section 4.2. Here, we cannot guarantee that the matrix $A - FX^i$ is stable. However, a stabilising initial guess on each level should be available from the next coarser grid.

Theorem 3 *Let X^1 denote an approximation to the solution X of the Riccati equation, $\|X^1 - X\| \leq \varepsilon$. Then the next iterate of nonlinear Richardson (17) with damping parameter λ is the same as the next iterate of linear Richardson (10) applied to the Lyapunov operator $\mathcal{L}_{X^1}$ (15) appearing in the Newton-Kleinman iteration (Algorithm 2). The first $\mathcal{O}(1)$ iterates Y^i of linear Richardson and X^i of nonlinear Richardson fulfil*

$$\|Y^i - X^1\| \leq \varepsilon \quad \Rightarrow \quad \|Y^i - X^i\| \leq \varepsilon^2.$$

Proof: The linearised system at X^1 is

$$\mathcal{L}_{X^1}(X) := (A^T - X^1F)X + X(A - FX^1) = -C - X^1FX^1.$$

The next iterate of linear Richardson fulfils

$$\begin{aligned} S(X^1, -C - X^1FX^1) &= X^1 + \lambda(\mathcal{L}_{X^1}(X^1) + C + X^1FX^1) \\ &= X^1 + \lambda(A^TX^1 + X^1A - X^1FX^1 - X^1FX^1 + C + X^1FX^1) \\ &= X^1 + \lambda(A^TX^1 + X^1A - X^1FX^1 + C) \\ &= \mathfrak{S}(X^1, -C). \end{aligned}$$

By X^i we denote the further iterates of nonlinear Richardson and by Y^i those of linear Richardson. Let $\|X^i - Y^i\| = \mathcal{O}(\varepsilon^2)$ and $\|X^1 - Y^i\| = \mathcal{O}(\varepsilon)$. Then the damping parameters λ_S used for S and $\lambda_{\mathfrak{S}}$ used for $\mathfrak{S}$ differ by $\|(X^i - Y^i)F\| = \mathcal{O}(\varepsilon^2)$. The difference of the iterates is thus

$$\begin{aligned} & \big(\mathcal{L}_{X^1}(Y^i) + C + X^1FX^1\big) - \big(\mathcal{R}(X^i) + C\big) \\ = \; & A^TY^i + Y^iA - X^1FY^i - Y^iFX^1 + C + X^1FX^1 - A^TX^i - X^iA + X^iFX^i - C \\ = \; & \underbrace{A^T(Y^i - X^i) + (Y^i - X^i)A}_{=\mathcal{O}(\varepsilon^2)} + \underbrace{(X^1 - Y^i)F(X^1 - Y^i)}_{=\mathcal{O}(\varepsilon^2)} - \underbrace{Y^iFY^i + X^iFX^i}_{=\mathcal{O}(\varepsilon^2)} \\ = \; & \mathcal{O}(\varepsilon^2). \end{aligned}$$

■

The previous theorem proves that one step of nonlinear Richardson is exactly linear Richardson applied to the system linearised at the current iterate. Thus, nonlinear Richardson should behave like Newton-Kleinman where each Newton step is not solved exactly but only by a single linear Richardson step.

5.4 Nonlinear Coarse Grid Correction

The idea of the coarse grid correction is as follows. Assume we have computed an approximate iterate X_ℓ^i on level ℓ so that the error

$$E_\ell^i := X_\ell - X_\ell^i, \qquad \mathcal{R}_\ell(X_\ell) := A_\ell^TX_\ell + X_\ell A_\ell - X_\ell F_\ell X_\ell = -C_\ell,$$

is fairly smooth (because of the ν_1 presmoothing steps). The error E_ℓ^i fulfills the equation

$$\mathcal{R}_\ell(X_\ell^i + E_\ell^i) = -C_\ell.$$

If $\mathcal{R}_\ell$ were linear, then we could just solve the defect equation $\mathcal{R}_\ell(E_\ell^i) = -\mathcal{R}_\ell(X_\ell^i) - C_\ell$ and update the iterate X_ℓ^i by the approximate solution $\tilde{E}_\ell^i$ of the defect equation. Since this is not reasonable

for the nonlinear operator $\mathcal{R}_\ell$, we have to use a different coarse-grid correction (cf. [33] for the general defect correction approach).

Let $X_{\ell-1}^{cg}$ denote a stabilising approximation to the solution $X_{\ell-1}$ on level $\ell-1$ (need not be highly accurate but stabilising). We define the (coarse) approximation

$$\hat{X}_\ell^{cg} := P_{\ell\leftarrow\ell-1}(X_{\ell-1}^{cg})$$

as the prolongation of a coarse-grid vector and the coarse right-hand side

$$C_{\ell-1}^{cg} := -\mathcal{R}_{\ell-1}(X_{\ell-1}^{cg}).$$

Then the error E_ℓ^i fulfils

$$\begin{aligned}\mathcal{R}_\ell(\hat{X}_\ell^{cg}+E_\ell^i) &= \mathcal{R}_\ell(\hat{X}_\ell^{cg})+\mathcal{R}_\ell(E_\ell^i)-\hat{X}_\ell^{cg}FE_\ell^i-E_\ell^iF\hat{X}_\ell^{cg}\\ &= \mathcal{R}_\ell(X_\ell^i+E_\ell^i)-\mathcal{R}_\ell(X_\ell^i)+\mathcal{R}_\ell(\hat{X}_\ell^{cg})+(X_\ell^i-\hat{X}_\ell^{cg})FE_\ell^i+E_\ell^iF(X_\ell^i-\hat{X}_\ell^{cg})\\ &= -C_\ell-\mathcal{R}_\ell(X_\ell^i)+\mathcal{R}_\ell(\hat{X}_\ell^{cg}) \quad +(X_\ell^i-\hat{X}_\ell^{cg})FE_\ell^i+E_\ell^iF(X_\ell^i-\hat{X}_\ell^{cg}).\end{aligned}$$

For $\varepsilon := \|E_\ell^i\|$ and $\delta := \|X_\ell^i-\hat{X}_\ell^{cg}\|$ we get the equation

$$\mathcal{R}_\ell(\hat{X}_\ell^{cg}+E_\ell^i) = -C_\ell-\mathcal{R}_\ell(X_\ell^i)+\mathcal{R}_\ell(\hat{X}_\ell^{cg})+\mathcal{O}(\delta\varepsilon)$$

where $D_\ell^i := C_\ell+\mathcal{R}_\ell(X_\ell^i)$ is the computable defect of the iterate X_ℓ^i. Since both $\hat{X}_\ell^{cg}$ and E_ℓ^i can be well represented on the coarse grid $\ell-1$, we can solve the defect equation on the coarse grid

$$\mathcal{R}_{\ell-1}(X_{\ell-1}^{cg}+E_{\ell-1}^i) = -R_{\ell-1\leftarrow\ell}(D_\ell^i)-C_{\ell-1}^{cg} \tag{19}$$

and obtain the approximation of E_ℓ^i by

$$E_\ell^i \approx P_{\ell\leftarrow\ell-1}(E_{\ell-1}^i).$$

The corrected iterate is

$$X_\ell^i := X_\ell^i + P_{\ell\leftarrow\ell-1}(E_{\ell-1}^i).$$

The coarse grid equation (19) will again be solved by the multigrid method where the starting value is $X_{\ell-1}^{cg}$.

The full V-cycle multigrid algorithm on level ℓ based on coarse grid approximations $(X_\iota^{cg})_{\iota=1}^{\ell-1}$ with corresponding right-hand sides $(C_\iota^{cg})_{\iota=1}^{\ell-1}$ and an initial approximation X_ℓ is given in Algorithm 4. There we perform ν_1 nonlinear Richardson steps before the coarse-grid correction (presmoothing) and ν_2 nonlinear Richardson steps after the coarse-grid correction (postsmoothing). The coarse grid equation is approximately solved by γ steps of nonlinear multigrid (typically $\gamma = 1$).

5.5 Positivity preservation

The Newton-Kleinman iteration guarantees that in each step the matrix $A-FY$ (Y being the stabilising iterate from the previous Newton step) is stable [26]. On the fine-grid $\ell = \ell_{\max}$ the right-hand side $C_\ell + Y_\ell F_\ell Y_\ell$ is symmetric positive semidefinite (SPSD) so that the solution X_ℓ is also SPSD. Due to roundoff errors or the error from the truncation to lower rank, it might happen that the approximate solution is only approximately SPSD. In order to avoid any instability due to this we can simply modify the truncation operator to project into the set of SPSD matrices.

Algorithm 4 Nonlinear Low-Rank Riccati-Multigrid Step

Procedure RMG(C_ℓ, ν_1, ν_2, γ, **var** X_ℓ)
{ $\mathcal{R}(X_\ell^{\mathrm{cg}}) = -C_\ell^{\mathrm{cg}}$ holds for the coarse-grid approximations X_ℓ^{cg} of X }
if $\ell = 1$ **then**
  Solve $\mathcal{R}(X_\ell) = -C_\ell$ {coarsest grid solve}
else
  for $j = 1, \ldots, \nu_1$ **do**
    $X_\ell := \mathcal{T}^{k \leftarrow 2k+k_C}\left(\mathfrak{S}(X_\ell, C_\ell)\right)$ { ν_1 presmoothing steps }
  end for
  $D_\ell := \mathcal{R}(X_\ell) + C_\ell$ { compute the defect }
  $C_{\ell-1} := \mathcal{T}^{k_C \leftarrow 2k+k_C}\left(-C_{\ell-1}^{\mathrm{cg}} - R_{\ell-1 \leftarrow \ell}(D_\ell)\right)$ { compute the coarse-grid right-hand side }
  $X_{\ell-1} := X_{\ell-1}^{\mathrm{cg}}$
  for $j = 1, \ldots, \gamma$ **do**
    Call RMG($C_{\ell-1}$, ν_1, ν_2, γ, $X_{\ell-1}$) {recursive call}
  end for
  $X_\ell := \mathcal{T}^{k \leftarrow 3k}\left(X_\ell + P_{\ell \leftarrow \ell-1}(X_{\ell-1} + X_{\ell-1}^{\mathrm{cg}})\right)$ { coarse-grid correction }
  for $j = 1, \ldots, \nu_2$ **do**
    $X_\ell := \mathcal{T}^{k \leftarrow 2k+k_C}\left(\mathfrak{S}(X_\ell, C_\ell)\right)$ { ν_2 postsmoothing steps }
  end for
end if

Let $R = UV^T$ be a symmetric $n \times n$ $R(k)$-matrix. Let $U = Q_U R_U$ be a QR decompositions with $Q_U \in \mathbb{R}^{n \times k}$ and $R_U \in \mathbb{R}^{k \times k}$. Then

$$R = UV^T = Q_U R_U V^T \overset{symm.}{=} Q_U R_U V^T Q_U Q_U^T$$

and thus the matrix $R_U V^T Q_U$ is symmetric. Let

$$R_U V^T Q_U = \tilde{U} \Lambda \tilde{U}^T, \qquad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_k),$$

be the eigenvalue decomposition of $R_U V^T Q_U$ with eigenvalues λ_i in descending order. Then for any $k' \in \{1, \ldots, k\}$ the matrix

$$\tilde{R} := \left(Q_U \tilde{U} \tilde{\Lambda}\right)\left(Q_U \tilde{U}\right)^T, \qquad \tilde{\Lambda} := \mathrm{diag}(\lambda_1, \ldots, \lambda_{k'}, 0, \ldots, 0)$$

is a rank k' best approximation of R in the set of SPSD matrices. It can be computed in $\mathcal{O}(nk^2)$ by the steps described above. If only symmetry but no positivity is needed then one can take the largest eigenvalues in modulus for the definition of $\tilde{\Lambda}$.

The SPSD truncation can be used for the fine-grid level $\ell = \ell_{\max}$, but on the coarse-grid levels the defect is not definite so that the solution is not SPSD. Still, symmetry can be preserved by the symmetric truncation.

For the fully nonlinear multigrid algorithm based on the nonlinear Richardson iteration the SPSD truncation on the fine-grid level $\ell = \ell_{max}$ is helpful, because the Richardson iteration does per se not guarantee that the positivity is preserved.

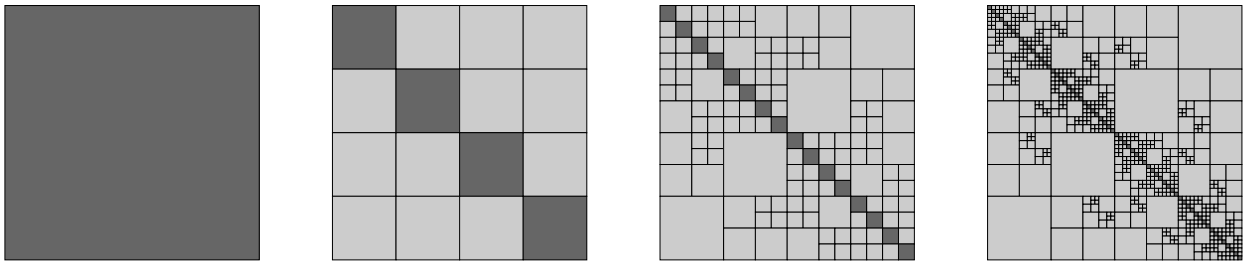

Figure 5: The $\mathcal{H}$-matrix format consists blockwise of $R(k)$-matrices (light grey blocks) and small full matrices (dark grey blocks). On level 1 (left) there is only a single full matrix block, whereas on level 4 (right) there are many $R(k)$-matrix blocks of varying size.

6 Hierarchical Matrices

A straight-forward generalisation of the low rank format is the blockwise low rank format. When the blocks are organised in a hierarchical way so that standard matrix arithmetics (addition, multiplication, inversion) are possible in almost optimal complexity, then we call such a matrix *hierarchical* matrix or short $\mathcal{H}$-matrix [17, 22], cf. Figure 5. The precise definition of an $\mathcal{H}$-matrix is given in the following.

Let $J := \{1, \dots, n\}$ denote the set of indices with corresponding basis functions $(\varphi_j)_{j\in J}$ defined on some grid used in the discretisation of the domain $\Omega \subset \mathbb{R}^d$. The so-called *cluster tree* is a multilevel partition of the index set J so that indices of corresponding geometrically connected basis functions are grouped together [17]. The formal definition is as follows.

Definition 4 (Cluster tree) *Let $J = \{1, \dots, n\}$ be an index set with n elements. A tree $T_J = (V_J, E_J)$ with vertices $V_J \ni v \subset J$, $v \neq \emptyset$, and $\mathrm{root}(T_J) = J$ is called a cluster tree of J, if*

$$\forall v \in V_J : \quad \{w \in V_J \mid (v,w) \in E_J\} = \emptyset \quad \text{or} \quad v = \dot{\bigcup_{\{w\in V_J \mid (v,w)\in E_J\}}} w.$$

Example 1 *Let $p \in \mathbb{N}$ and $n = 2^p$ be the cardinality of the index set $J = \{1, \dots, n\}$. We define a cluster tree T_J of J as follows: the root of the tree is J. The successors of J are $\{1, \dots, n/2\}$ and $\{n/2 + 1, \dots, n\}$. Each of the vertices v has the form $v = \{j2^i + 1, \dots, (j + 1)2^i\}$ for some $i \in \{0, \dots, p\}$ and $j \in \{0, \dots, 2^{p-i} - 1\}$ and the successors s_1, s_2 of such a vertex are (see Figure 6)*

$$s_1 := \{j2^i + 1, \dots, (j + 1/2)2^i\}, \qquad s_2 := \{(j + 1/2)2^i + 1, \dots, (j + 1)2^i\}.$$

The cluster tree constructed in this way is a binary tree, which means that each vertex is either a leaf or has exactly two successors.

The cluster tree defines candidates for blocks $v \times w \subset J \times J$ so that the corresponding matrix block $M|_{v\times w}$ for a matrix $M \in \mathbb{R}^{J\times J}$ can be approximated by low rank. The trivial tree would be of cardinality one where the root is the only node and $J \times J$ the only candidate for a low rank block. In this case we are back at the $R(k)$-matrices discussed in the previous sections. Blocks $v \times w$ that

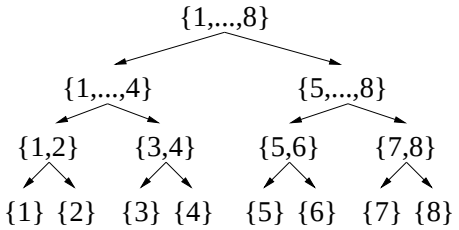

Figure 6: The cluster tree T_J: the index set $J = \{1, \ldots, 8\}$ is successively partitioned.

are suitable for low rank approximation are characterised by the so-called *admissibility condition*

$$\min\{\mathrm{diam}(\Omega_v), \mathrm{diam}(\Omega_w)\} \leq \mathrm{dist}(\Omega_v, \Omega_w), \qquad \Omega_v := \bigcup_{j \in v} \mathrm{supp}\varphi_j \subset \Omega. \tag{20}$$

The cluster tree T_J together with the admissibility condition (20) defines the block cluster tree whose leaves will yield the partition of the matrix into low rank blocks.

Definition 5 (Block cluster tree) *A block cluster tree $T = (V, E)$ corresponding to a cluster tree $T_J = (V_J, E_J)$ is a cluster tree of the index set $J \times J$ where each vertex $t \in V$ is of the form $t = v \times w$ with vertices $v, w \in V_J$. Each leaf $t \in V$ is either admissible, i.e. v and w fulfil (20), or one of the two clusters v or w is a leaf of the cluster tree T_J.*

The construction of the block cluster tree is straight-forward. We start with the root $J \times J$ and further on define the sons of a node $v \times w \in V$ by either the empty set (if it is admissible) or by the product of the sons of v and w. The leaves of the block cluster tree yield a partition and this partition is used to define the block structure of an $\mathcal{H}$-matrix.

Definition 6 ($\mathcal{H}$-matrix) *Let $T = (V, E)$ be a block cluster tree (index set J) and $k \in \mathbb{N}$. A matrix $M \in \mathbb{R}^{J \times J}$ is said to be an $\mathcal{H}$-matrix corresponding to T with blockwise rank k if for all leaves (vertex with no successor) $v \times w \in V$ the submatrix $M|_{v \times w}$ fulfils* $\mathrm{rank}(M|_{v \times w}) \leq k$.

An $\mathcal{H}$-matrix is stored blockwise in $R(k)$-matrix format. The storage requirements are $\mathcal{O}(kn \log n)$ and the matrix by vector multiplication can be performed in $\mathcal{O}(kn \log n)$ [17]. However, the set of $\mathcal{H}$-matrices with fixed blockwise rank limit k is not closed with respect to addition. In analogy to the $R(k)$-matrix truncation operator we define the $\mathcal{H}$-matrix truncation $\mathcal{T}_{\mathcal{H}}^{k \leftarrow k'}$ by applying in each admissible block the $R(k)$-matrix truncation $\mathcal{T}^{k \leftarrow k'}$. The formatted addition $\oplus$ is then defined by

$$A \oplus B := \mathcal{T}_{\mathcal{H}}^{k \leftarrow 2k}(A + B).$$

The formatted addition of two $\mathcal{H}$-matrices is of complexity $\mathcal{O}(k^2 n \log n)$[17].

6.1 Existence of $\mathcal{H}$-matrix solutions

Theorem 1 can be generalised as follows. Let C be an $\mathcal{H}$-matrix of the format defined in [15] or [18] (depicted in Figure 5) instead of a low rank matrix. The matrix A is the finite difference or

finite element stiffness matrix and F is of rank $k_F = \mathcal{O}(1)$. Let k_C denote the blockwise rank of C. Then the solution X to (1) can be approximated by an $\mathcal{H}$-matrix $\tilde{X}$ with blockwise rank

$$k_{\tilde{X}} = k_\varepsilon(k_C + k_F), \qquad k_\varepsilon = \mathcal{O}(\log(1/\varepsilon)^\delta),$$

up to a relative error of ε, where $\delta = 2$ for the $\mathcal{H}$-matrix format from [15] and $\delta = 6$ for the $\mathcal{H}$-matrix format from [18]. In practice one observes the same dependency of $k_{\tilde{X}}$ on the spectrum of A and the approximation quality ε as for the low rank case.

6.2 Nonlinear $\mathcal{H}$-matrix Richardson

The (formatted) Richardson iteration (16) is also applicable for $\mathcal{H}$-matrices C and corresponding $\mathcal{H}$-matrix solutions X. The quadratic low rank term $XFX = X\sum_{i=1}^{k_F} u_i v_i^T X = \sum_{i=1}^{k_F}(Xu_i)(X^T v_i)^T$ is blockwise given in $R(k_F)$-matrix representation and therefore compatible with the format of X and C. The matrices $A^T X$ and XA can be converted to the $\mathcal{H}$-matrix format by the techniques from [17]. Even non-local $\mathcal{H}$-matrices A that stem, e.g., from some boundary element discretisation, can be treated in this way. For a (local) sparse matrix A the complexity of the matrix multiplications $A^T X$ and XA is $\mathcal{O}(k^2 n \log n)$ so that the total complexity of one Richardson step is $\mathcal{O}(k^2 n \log n)$.

6.3 Prolongation and Restriction of $\mathcal{H}$-matrices

The matrix prolongation and restriction are geometrically local operations and can be treated in the same way as the sparse system matrix A, i.e., the prolongation of an $\mathcal{H}$-matrix $M_{\ell-1}$ from level $\ell - 1$ to level ℓ can be done in $\mathcal{O}(k^2 n \log n)$. This can even be simplified using the following trick: Let $M_{\ell-1}|_{v_{\ell-1}\times w_{\ell-1}}$ be some matrix block of $M_{\ell-1}$ in $R(k)$-matrix format. The corresponding fine-grid block is $M_\ell|_{v_\ell \times w_\ell}$. Instead of the interpolation of values at fine-grid nodes between coarse-grid nodes, we use the interpolation only for the interior nodes of $v_{\ell-1}, w_{\ell-1}$ and extrapolate the values at the boundary. Thereby, the prolongation avoids to combine entries from different blocks and thus the $R(k)$-matrix format is retained as in the global low-rank case (no truncation necessary). The restriction is again the adjoint of the prolongation. The simplified matrix prolongation and restriction is of optimal complexity $\mathcal{O}(kn \log n)$ (because the truncation is avoided).

7 NUMERICAL EXAMPLES

The numerical tests in this Section serve three purposes: first, we investigate the influence of the nonlinearity XFX that can be steered by the parameter κ in our model problem (3). Second, we use the low-rank format in the multigrid method and observe the dependency of the convergence rate on the truncation rank. Third, we compare the fully nonlinear multigrid (Algorithm 4) with the Newton-multigrid (Algorithm 3) and the Newton-Kleinman iteration based on linear multigrid (Algorithms 2 and 1) for large-scale Riccati equations. At last we shall compare the Cholesky factor ADI iteration [21] to the linear multigrid method (Algorithm 1) for the symmetric Lyapunov equation.

7.1 Influence of the Nonlinearity

We consider the model problem from Section 2.3 for different parameters κ used in the definition of the matrix F of the quadratic term in the Riccati equation (1). We fix the level $\ell := 6$ with $n_\ell = 16129$ interior grid-points and the parameter $\beta := 20$ for the non-symmetry of the matrix A. In the multigrid Algorithm 4 we use $\nu_1 := 2$ presmoothing, $\nu_2 := 1$ postsmoothing and $\gamma := 1$ coarse-grid steps. The error $\varepsilon(i) := \|X^i - X\|_2 / \|X\|_2$ and convergence rates $\|X^i - X\|_2 / \|X^{i-1} - X\|_2$ during the

	$\kappa = 1$		$\kappa = 100$		$\kappa = 10000$		$\kappa = 1000000$	
i	ε	*rate*	ε	*rate*	ε	*rate*	ε	*rate*
0	3.1×10^{-2}		3.0×10^{-2}		4.6×10^{-2}		5.4×10^{-2}	
1	1.0×10^{-2}	0.32	1.0×10^{-2}	0.34	1.4×10^{-2}	0.30	3.8×10^{-3}	0.07
2	3.8×10^{-3}	0.38	3.7×10^{-3}	0.37	3.9×10^{-3}	0.29	6.1×10^{-4}	0.16
3	1.3×10^{-3}	0.34	1.3×10^{-3}	0.35	1.1×10^{-3}	0.29	4.1×10^{-4}	0.68
4	7.8×10^{-4}	0.60	7.4×10^{-4}	0.58	3.5×10^{-4}	0.31	3.2×10^{-4}	0.78
5	3.5×10^{-4}	0.45	3.2×10^{-4}	0.43	1.2×10^{-4}	0.35	2.5×10^{-4}	0.80
6	1.5×10^{-4}	0.41	1.3×10^{-4}	0.39	7.2×10^{-5}	0.59	1.9×10^{-4}	0.76

Table 3: The relative error ε of the iterate X^i after $i = 0, \ldots, 6$ nonlinear multigrid iterations.

first six iterations are contained in Table 3. As a reference solution X we compute a rank 30 solution using 50 Newton-multigrid steps (Algorithm 3) so that the relative defect $\|\mathcal{R}(X) - C\|/\|C\|$ of the symmetric positive semidefinite solution (cf. Section 5.5) is less than 5×10^{-12} for $\kappa = 1, 100, 10000$ (the relative approximation error should be of similar size).

For very large values of κ the convergence rate tends to 1. However, in the first two steps the convergence rate is always much smaller so that the discretisation error is met even for large κ after two steps. We conclude that the nonlinear multigrid method is suitable even for large κ, i.e. a large scaling of the nonlinearity.

7.2 Comparison between Newton-Kleinman and Nonlinear Multigrid

The Newton-Kleinman iteration (with initial guess $X := 0$) suffers from the increasing dominance of the nonlinearity. On the fine-grid the error is roughly halved in each Newton step so that many steps are necessary to find an approximation with accuracy of the size of the discretisation error. The number of multigrid steps for the solution of the linear Lyapunov equation in each step is completely irrelevant.

The Newton-multigrid Algorithm 3 from Section 5.1 can overcome the first obstacle, namely the bad initial guess, by using the prolonged coarse-grid solution as an initial guess. Therefore, the number of Newton steps is reduced. In principle we would have to solve the linear systems in each Newton step exactly, or at least highly accurate. In the numerical tests we observe that the number of multigrid steps necessary in each Newton step is just one. This is much less than the number of steps required for an almost exact solve. Using just a single multigrid step yielded the best results for the Newton-multigrid which are summarised in Table 4. We observe that the convergence of Newton-multigrid is quite similar to that of the fully nonlinear multigrid. A Newton-multigrid step is slightly more expensive than a nonlinear multigrid step. However, both methods yield the

	$\kappa = 1$		$\kappa = 100$		$\kappa = 10000$		$\kappa = 1000000$	
i	ε_X	*rate*	ε_X	*rate*	ε_X	*rate*	ε_X	*rate*
0	3.1×10^{-2}		3.0×10^{-2}		4.6×10^{-2}		5.4×10^{-2}	
1	1.0×10^{-2}	0.32	1.0×10^{-2}	0.34	2.4×10^{-2}	0.52	4.4×10^{-3}	0.08
2	3.6×10^{-3}	0.37	3.6×10^{-3}	0.35	6.0×10^{-3}	0.25	1.5×10^{-3}	0.34
3	1.3×10^{-3}	0.35	1.3×10^{-3}	0.35	2.2×10^{-3}	0.36	1.4×10^{-3}	0.89
4	7.5×10^{-4}	0.59	7.4×10^{-4}	0.59	1.5×10^{-3}	0.67	1.2×10^{-3}	0.90
5	4.4×10^{-4}	0.59	4.2×10^{-4}	0.57	1.2×10^{-3}	0.84	1.1×10^{-3}	0.92
6	2.7×10^{-4}	0.60	2.4×10^{-4}	0.57	9.5×10^{-4}	0.68	1.1×10^{-3}	0.93

Table 4: The relative error ε_X of the iterate X^i after $i = 0, \ldots, 5$ Newton-multigrid iterations.

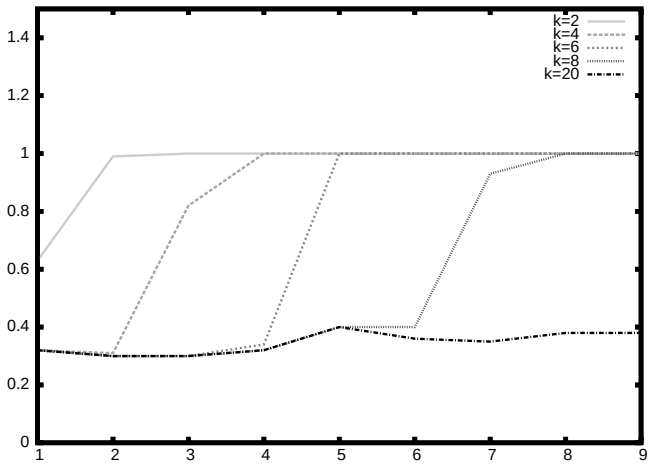

Figure 7: Convergence rates after $i = 1, \ldots, 9$ iterations on level $\ell = 6$ using a rank $k = 2, 4, 6, 8, 20$.

solution after a few steps in $\mathcal{O}(n)$ complexity.

7.3 Influence of the Truncation

In this section we set the parameter $\kappa := 1000$ and $\beta = 20$, so that the nonlinearity is well pronounced, and we vary the rank k for the representation of the iterates X_ℓ^i between 2, 4, 6 and 20 on level $\ell := 6$. Obviously, a rank of $k = 0$ will yield the trivial approximation $X_\ell = 0$, whereas the full rank $k = 16129$ will yield the exact solution.

The reference solution is — as in the previous section — a rank 30 approximation computed by 50 multigrid steps so that the relative residual is less than 5×10^{-12}. In Figure 7 we depict the convergence rates of the relative error $\varepsilon_X := \|X_\ell^i - X_\ell\|_2 / \|X_\ell\|_2$ for the first nine iterates X^i using a truncated representation of rank k in Algorithm 4. For $k = 20$ we can see the standard convergence rate, just as if we had performed the iteration without any truncation at all. For lower rank $k = 2, 4, 6$ the convergence breaks down (rate 1.0) as the relative error approaches the relative error of a best approximation of rank k. The transition is quite sharp in the sense that only one or two steps during the multigrid iteration have a convergence rate larger than the standard one and less

than 1.0.

7.4 Influence of the Non-symmetry of A

The non-symmetry of the stiffness matrix A has a strong effect on the convergence of the multigrid method when it is used as a solver for systems $Ax = b$. The same behaviour is to be expected when we apply the (nonlinear) multigrid method for the Riccati operator. We test this by varying the parameter β in the setup of the stiffness matrix A between $\beta = 20$ and $\beta = 160$. Due to the fact that in the partial differential equation (3) the term $2\beta\partial_{\xi_2}x(t,\xi)$ is of lower order, we have to reach a certain minimal level $\ell_{\min}$ from which on the term will be harmless, but as ℓ tends to 1 the convergence of Richardson as well as the multigrid iteration will tend to 1 or above. This means, depending on the parameter β, there is a lower bound $\ell_{\min}$ on the allowable coarse grid where we have to solve the system by some other means than multigrid. Here, we apply sufficiently many steps of the smoother on the coarse grid, but the ADI solver from Section 7.6 (for non-symmetric systems) would also be a good choice. The convergence rates for the nonlinear multigrid Algorithm 4 on level $\ell = 6$ with optimal choice of the coarse-grid $\ell_{\min}$ are contained in Table 5.

	$\beta = 20$		$\beta = 40$		$\beta = 80$		$\beta = 160$	
i	ε	*rate*	ε	*rate*	ε	*rate*	ε	*rate*
0	3.2×10^{-2}		3.7×10^{-2}		5.3×10^{-2}		8.7×10^{-2}	
1	1.0×10^{-2}	0.32	1.2×10^{-2}	0.33	2.0×10^{-2}	0.37	3.5×10^{-2}	0.41
2	3.1×10^{-3}	0.30	4.8×10^{-3}	0.39	8.8×10^{-3}	0.45	1.5×10^{-2}	0.43
3	9.4×10^{-4}	0.30	1.4×10^{-3}	0.28	3.2×10^{-3}	0.37	7.6×10^{-3}	0.50
4	3.0×10^{-4}	0.32	5.2×10^{-4}	0.39	2.1×10^{-3}	0.64	3.2×10^{-3}	0.43
5	1.2×10^{-4}	0.40	2.0×10^{-4}	0.39	7.9×10^{-4}	0.38	1.8×10^{-3}	0.57
6	4.3×10^{-5}	0.36	8.3×10^{-5}	0.41	5.9×10^{-4}	0.75	1.3×10^{-3}	0.69

Table 5: The relative error ε of the iterate X_ℓ^i after $i \in \{1,\dots,6\}$ iterations using a rank $k = 10$ for the approximation of the iterates in the $R(k)$-matrix format.

7.5 Large Scale Riccati Equations

For large scale Riccati equations there are two important questions to be answered. First, is the convergence rate bounded away from one independently of the level, and second, how does the method scale in n?

For the first question we depict the convergence rates for the first nine iterates of Algorithm 4 on level $\ell = 5, 6, 7, 8$ in Figure 8. The rank $k = 20$ is used to represent the iterates X_ℓ^i and we can see that the convergence rate seems to stabilise as ℓ grows. On level $\ell = 8$ the rate stays below 0.4. Moreover, the convergence rate in the first two steps tends to be less than 0.3 for larger level numbers (we have used a more conservative damping factor in the Richardson iteration on level 1–3 which influences the convergence rate on smaller levels). In the nested iteration it will therefore be sufficient to apply two multigrid steps per level and on the final level two multigrid steps are sufficient so that the relative approximation error of the discrete solution is less than 50% of the discretisation error.

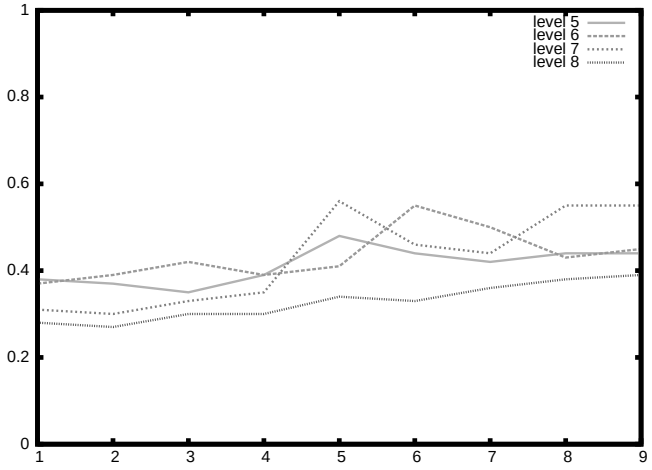

Figure 8: Convergence rates after $i = 1, \ldots, 9$ iterations on level $\ell = 5, 6, 7, 8$.

The answer to the second question is given in Table 6 where we used the nested iteration based on the nonlinear Riccati-multigrid Algorithm 4 with two steps on the final level. The rank for the representation of the (approximate) solution X_ℓ is increasing with increasing level number in order

	$\ell = 6, k = 3$	$\ell = 7, k = 4$	$\ell = 8, k = 5$	$\ell = 9, k = 6$	$\ell = 10, k = 7$
	$n_6 = 16129$	$n_7 = 65025$	$n_8 = 261121$	$n_9 = 1046529$	$n_{10} = 4190209$
$i = 0$	3.3×10^{-2}	1.5×10^{-2}	6.2×10^{-3}	2.9×10^{-3}	1.2×10^{-3}
$i = 1$	1.2×10^{-2}	5.1×10^{-3}	1.7×10^{-3}	7.5×10^{-4}	2.8×10^{-4}
$i = 2$	7.1×10^{-3}	2.7×10^{-3}	9.1×10^{-4}	3.4×10^{-4}	1.3×10^{-4}
Time (Sec.)	1.9	11.5	90	956	6376

Table 6: The table quotes the relative error ε of the iterate X_ℓ^i after i iterations on level ℓ using a rank k for the approximation of the iterates. The last row reports the computing time in seconds.

to stay well below the discretisation error with the relative approximation error ε. The numerical tests were all performed on a SUN ULTRASPARC III with 900 MHz CPU clock rate and 150 MHz memory clock rate. We have used the standard BLAS and LAPACK implementations provided by the machine vendor. The algorithms are implemented in the *C* programming language.

7.6 Comparison between Cholesky Factor ADI and Linear Multigrid

The ADI iteration for the solution of a symmetric Lyapunov equation $AX + XA + C = 0$ reads

$$X^i = -2p_i(A + p_iI)^{-1}C(A + p_iI)^{-1} + (A + p_iI)^{-1}(A - p_iI)X^{i-1}(A - p_iI)(A + p_iI)^{-1},$$

where the shift parameters p_i have to be chosen appropriately [31]. As an initial guess we use $X^0 := 0$. The matrix A is the finite difference stiffness matrix $A_\ell \in \mathbb{R}^{n_\ell \times n_\ell}$ on level ℓ defined in Section 2. The symmetric positive semidefinite matrix C is of rank $k_C := 6$ and the exact solution X is of rank 3.

For the comparison with the multigrid method we use the Cholesky factor ADI (CF-ADI) algorithm [21, Algorithm 3] with (non-cyclic) shift parameters $p_1, \ldots, p_{J_{max}}$ as defined in [38]. Additionally, the iterates $X^i = Z_i^{cfadi} \left(Z_i^{cfadi}\right)^T$ will be truncated in each step:

$$X^i \quad \leftarrow \quad \mathcal{T}^{k \leftarrow k+k_C}(X^i).$$

Without the truncation in each step the final rank k of X^J after J ADI steps would be $k = Jk_C$ and a truncation to minimal rank would be rather expensive.

In each step of the CF-ADI iteration we have to invert systems of the form $A + p_i I$. For this, we use the $\mathcal{H}$-matrix Cholesky factorisation based on a nested dissection clustering technique [19]. The factorisation of the matrix $A + p_i I$ (single precision accuracy) on level $\ell = 8$ with $n_\ell = 261121$ degrees of freedom takes less than 36 seconds and uses less than 240MB main memory. Once the factorisation is computed, one can solve the system for a given right-hand side in 2.2 seconds via forward/backward substitution. These results are comparable to those obtained by SuperLU [10]. The factorisations account for roughly 2/3 of the time of the CF-ADI iteration. In Table 7 we report the time used for the CF-ADI iteration and for the (linear) multigrid iteration on level $\ell = 6, 7, 8$. We conclude that the multigrid method is by a factor of 10 faster, but in principle both

	$\ell = 6,\ \varepsilon = 3{\times}10^{-3}$	$\ell = 7,\ \varepsilon = 1{\times}10^{-3}$	$\ell = 8,\ \varepsilon = 3{\times}10^{-4}$
CF-ADI	26.9	151.7	1037
Multigrid	2.7	12.3	74.3

Table 7: The table quotes the time required to compute an approximate solution with relative error ε on level $\ell = 6, 7, 8$ by the CF-ADI and the multigrid algorithm.

methods scale almost linearly.

References

[1] Absil PA, Sepulchre R, Van Dooren P, Mahony R: *Cubically convergent iterations for invariant subspace computation.* SIAM Journal on Matrix Analysis and Applications 26, 70–96, 2004.

[2] Antoulas A, Sorensen D, Zhou Y: *On the decay rate of Hankel singular values and related issues.* Systems and Control Letters 46, 323–342, 2002.

[3] Arnold FW, Laub AJ: *Generalized eigenproblem algorithms and software of algebraic Riccati equations.* Proceedings of IEEE 72, 1746–1754, 1984.

[4] Banks H, Kunisch K: *The linear regulator problem for parabolic systems.* SIAM Journal on Control and Optimization. 22, 684–696, 1984.

[5] Bartels RH, Stewart GW: *Solution of the matrix equation $AX + XB = C$.* Communications of the Association for Computing Machinery 15, 820–826, 1972.

[6] Benner P, Byers R, Quintana-Orti E, Quintana-Orti G: *Solving algebraic Riccati equations on parallel computers using Newton's method with exact line search.* Parallel Computing 26, 1345–1368, 2000.

[7] Byers R: *Numerical condition of the algebraic Riccati equation.* Contemporary Mathematics 47, 35–49, 1985.

[8] Casti JL: *Dynamical systems and their applications: linear theory.* Academic Press, New York, 1977.

[9] Curtain R, Pritchard A: *Infinite dimensional linear systems theory.* Springer-Verlag, New York, 1978.

[10] Demmel JW, Eisenstat SC, Gilbert JR, Li XS, Liu JWH: *A supernodal approach to sparse partial pivoting.* SIAM Journal on Matrix Analysis and Applications 20, 720–755, 1999.

[11] Gantmacher FR: *Theory of Matrices.* American Mathematical Society, New York, 2001.

[12] Gibson JS: *The Riccati integral equations for optimal control problems on Hilbert spaces.* SIAM Journal on Control and Optimization 17, 537–565, 1979.

[13] Golub GH, Nash S, Van Loan CF: *A Hessenberg-Schur method for the matrix problem* $AX + XB = C$. IEEE Transactions on Automatic Control 24, 909–913, 1979.

[14] Golub GH, Van Loan CH: *Matrix Computations.* Johns Hopkins University Press, London, 1996.

[15] Grasedyck L: *Existence of a low rank or* $\mathcal{H}$*-matrix approximant to the solution of a Sylvester equation.* Numerical Linear Algebra with Applications 11, 371–389, 2004.

[16] Grasedyck L, Hackbusch W: *A multigrid method to solve large scale Sylvester equations.* Technical Report 48, Max Planck Institute for Mathematics in the Sciences, Leipzig, 2004. http://www.mis.mpg.de/preprints/2004/prepr2004_48.html

[17] Grasedyck L, Hackbusch W: *Construction and arithmetics of* $\mathcal{H}$*-matrices.* Computing 70, 295–334, 2003.

[18] Grasedyck L, Hackbusch W, Khoromskij B: *Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices.* Computing 70, 121–165, 2003.

[19] Grasedyck L, Kriemann R, LeBorne S: *Parallel black box domain decomposition based* $\mathcal{H}$*-LU preconditioning.* Technical Report 115, Max Planck Institute for Mathematics in the Sciences, Leipzig, 2005. http://www.mis.mpg.de/preprints/2005/prepr2005_115.html

[20] Guo C, Lancaster P: *Analysis and modification of Newton's method for algebraic Riccati equations.* Mathematics of Computation 67, 1089–1105, 1998.

[21] Li JR, White J: *Low rank solution of Lyapunov equations.* SIAM Journal on Matrix Analysis and Applications. 24, 260–280, 2002.

[22] Hackbusch W: *A sparse matrix arithmetic based on* $\mathcal{H}$*-matrices. Part I: Introduction to* $\mathcal{H}$*-matrices.* Computing 62, 89–108, 1999.

[23] Hackbusch W: *Elliptic differential equations. Theory and numerical treatment.* Springer-Verlag, Berlin, 2nd edition, 2003.

[24] Hackbusch W: *Iterative solution of large sparse systems.* Springer-Verlag, Berlin, 2nd edition, 2003.

[25] Hackbusch W: *Multi-Grid Methods and Applications.* Springer-Verlag, Berlin, 2nd edition, 2003.

[26] Kleinman DL: *On an iterative technique for Riccati equations computation.* IEEE Transactions on Automatic Control 13, 114–115, 1968.

[27] Kwakernaak H, Sivan R: *Linear optimal control systems.* Wiley-Interscience New York, 1972.

[28] Lasiecka I, Triggiani R: *Control theory for partial differential equations: continuous and approximation theories.* Cambridge University Press, Cambridge, 2000.

[29] Laub A: *A Schur method for solving algebraic Riccati equations.* IEEE Transactions on Automatic Control 24, 913–921, 1979.

[30] Laub A: *Invariant subspace methods for the numerical solution of Riccati equations.* In S. Bittanti, A. Laub, and J. Willems (eds.): The Riccati equation, Springer-Verlag, Berlin, 163–196, 1991.

[31] Lu A, Wachspress L: *Solution of Lyapunov equations by alternating direction implicit iteration.* Computers and Mathematics with Applications 21, 43–58, 1991.

[32] Mehrmann VL: *The autonomous linear quadratic control problem, theory and numerical solution.* Lecture Notes in Control and Information Sciences 163, Springer-Verlag, Heidelberg, 1991.

[33] Mehrmann VL, Tan E: *Defect correction methods for the solution of algebraic Riccati equations.* IEEE Transactions on Automatic Control 33, 695–698, 1988.

[34] Samarskii AA, Nikolaev ES: *Numerical methods for grid equations. Vol. II: Iterative methods.* Birkhäuser, Basel, 1989.

[35] Lezius R, Tröltzsch F: *Theoretical and numerical aspects of controlled cooling of steel profiles.* In H. Neunzert (eds.): Progress in industrial mathematics at ECMI94, Wiley-Teubner, Leipzig, 380–388, 1996.

[36] Morris K, Navasca C: *Solution of algebraic Riccati equations arising in control of partial differential equations.* In Zolesio, J.P., and Cagnol, J., (eds.): Control and Boundary Analysis, Marcel Dekker, New York, 2004.

[37] Penzl T: *A multi-grid method for generalized Lyapunov equations.* Technical Report No. 24, SFB 393 at University Chemnitz, 1997.

[38] Penzl T: *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case.* Systems and Control Letters 40, 139–144, 2000.

[39] Penzl T: *A cyclic low rank Smith method for large sparse Lyapunov equations.* SIAM Journal on Scientific Computing 21, 1401–1418, 2000.

[40] Reusken A: *On maximum norm convergence of multigrid methods for two-point boundary value problems.* SIAM Journal on Numerical Analysis 29, 1569–1578, 1992.

[41] Roberts JD: *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function.* International Journal of Control 32, 677–687, 1980.

[42] Rosen JIG, Wang C: *A multilevel technique for the approximate solution of operator Lyapunov and algebraic Riccati equations.* SIAM Journal on Numerical Analysis 32, 514–541, 1995.

[43] Stéphanos C: *Sur une extension du calculus des substitutions linéaires.* Journal de Mathematiques Pures et Appliquees 6, 73–128, 1900.

[44] Stykel T: *Generalized Alternating Direction Implicit Method for Projected Generalized Lyapunov Equations.* Proceedings in Applied Mathematics and Mechanics 4, 686–687, 2004.

[45] Trottenberg U, Oosterlee C, Schüller A: *Multigrid*, Academic Press London, 2001.

[46] Van Dooren P: *A generalized eigenvalue approach for solving Riccati equations.* SIAM Journal on Scientific and Statistical Computing 2, 121–135, 1981.

[47] Yueh WC: *Eigenvalues of several tridiagonal matrices.* Applied Mathematics E-Notes 5, 66–74, 2005.