Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

A Multigrid Method to Solve Large Scale Sylvester Equations

(revised version: May 2007)

by

Lars Grasedyck, and Wolfgang Hackbusch

Preprint no.: 48 2004

A MULTIGRID METHOD TO SOLVE LARGE SCALE SYLVESTER EQUATIONS

LARS GRASEDYCK† AND WOLFGANG HACKBUSCH†

Abstract. We consider the Sylvester equation $AX - XB + C = 0$, where the matrix $C \in \mathbb{R}^{n\times m}$ is of low rank and the spectra of $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ are separated by a line. The solution X can be approximated in a data-sparse format and we develop a multigrid algorithm that computes the solution in this format. For the multigrid method to work we need a hierarchy of discretisations. Here, the matrices A and B each stem from the discretisation of a partial differential operator of elliptic type. The algorithm is of complexity $\mathcal{O}(n + m)$, or, more precisely, if the solution can be represented with $(n+m)k$ data ($k \sim \log(n+m)$) then the complexity of the algorithm is $\mathcal{O}((n+m)k^2)$.

Key words. Fast solver, Lyapunov equation, Riccati equation, Sylvester equation, control problem, low rank approximation, multigrid method

AMS subject classifications. 65F05, 65F30, 65F50

1. Introduction.

In this article we consider the matrix Sylvester equation

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

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{m\times m}, C \in \mathbb{R}^{n\times m}$ are given input matrices and the sought solution is $X \in \mathbb{R}^{n\times m}$. We make two assumptions concerning the matrices A, B, C.

First, A and $-B$ are stiffness matrices from the discretisation of a linear elliptic partial differential operator. This allows for the use of a multigrid method to solve the Sylvester equation in $\mathcal{O}(nm)$ for a general matrix C. In the outlook we comment on the case when A and B are general sparse matrices.

Second, the matrix C is of low rank k_C, i.e., given in factorised form $C = UV^T$ with matrices $U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{m\times k}$. Under these assumptions the solution X can be approximated by a matrix $\tilde{X}$ of rank $k = \mathcal{O}(|\log\varepsilon|k_C)$ such that $\|X - \tilde{X}\|_2 \le \varepsilon$. The multigrid method can be adapted so that it is of linear complexity $\mathcal{O}((n+m)k^2)$ instead of quadratic complexity.

In the following Section we will consider a simple model problem where the multigrid techniques are applicable without further complications. The model is an optimal control problem that leads to an algebraic matrix Riccati equation which can be solved iteratively by Newton's method so that in each step a Lyapunov equation $A^TX + XA = C$ has to be solved. Such a Lyapunov equation is a special case of the more general Sylvester equation. We also give an example from model reduction where the Sylvester equation appears directly for the computation of cross-Gramians. In Section 2 we give a short introduction to low rank arithmetics. Section 3 examines the tensor structure of a Sylvester equation and as a special case we consider diagonal Sylvester equations in Section 4. This special case is the basis for the Jacobi iteration introduced in Section 5. In Section 6 we derive the multigrid method and proof it's convergence. At last we present numerical results for large scale matrix equations.

1.1. Model Problem.

The model problem to be introduced in this Section is the (distributed) control of the two-dimensional heat equation (cf. [13] and the references therein) which is used, e.g., in optimal control problems for the selective cooling of steel [14]. The domain where the PDE is posed is the unit square. Using

†Max Planck Institute for Mathematics in the Sciences, Inselstr. 22-26, D-04103 Leipzig, Germany (lgr@mis.mpg.de, wh@mis.mpg.de)

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.

1.1.1. Continuous Model. We fix the domain $\Omega := (0,1) \times (0,1)$ and the boundary $\Gamma := \partial\Omega$. The goal is to minimise the quadratic performance index

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

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

$$\begin{array}{lcll} \partial_t x(t,\xi) & = & \partial^2_{\xi_1} x(t,\xi) + \partial^2_{\xi_2} x(t,\xi) + \kappa(\xi)u(t), & \xi \in \Omega, \quad t \in (0,\infty), \\ x(t,\xi) & = & 0, & \xi \in \Gamma, \quad t \in (0,\infty), \\ x(0,\xi) & = & x_0, & \xi \in \Omega, \\ y(t) & := & \int_\Omega \omega(\xi) x(t,\xi) d\xi, & t \in (0,\infty). \end{array} \tag{1.1}$$

The values of κ and ω are

$$\kappa(\xi) := \begin{cases} 1 & \xi \in (\frac{1}{2},1) \times (0,1), \\ 0 & \text{otherwise.} \end{cases}, \qquad \omega(\xi) := \begin{cases} 1 & \xi \in (0,1) \times (\frac{1}{2},1), \\ 0 & \text{otherwise.} \end{cases}.$$

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, 13].

1.1.2. Semidiscretisation by Finite Differences. The differential equation (1.1) is discretised by finite differences on a uniform mesh of $[0,1]^2$ with n interior grid-points $(x_i)_{i=1}^n$ and mesh width $h = (\sqrt{n}+1)^{-1}$. By ϕ_i we denote the piecewise linear interpolant on the mesh with $\phi_i(x_i) = 1$ and $\phi_i(x_j) = 0$ for $j \neq i$. The corresponding space-discrete system is

$$\begin{array}{lcll} \partial_t x(t) & = & Ax(t) + Ku(t), & t \in (0,\infty), \\ x(0) & = & x_0, & \\ y(t) & := & Wx(t), & t \in (0,\infty), \end{array} \tag{1.2}$$

where $A := A^{FD} \in \mathbb{R}^{n\times n}$ is the standard finite difference discretisation of the 2d Laplacian, $x(t) \in \mathbb{R}^n$, $u(t), y(t) \in \mathbb{R}$ and the vectors $K := K^{FD} \in \mathbb{R}^n$ and $W \in \mathbb{R}^n$ are

$$K_i^{FD} := \kappa(x_i), \qquad W_i := \int_\Omega \omega(\xi)\phi_i(\xi)\mathrm{d}\xi. \tag{1.3}$$

The stiffness matrix A is symmetric negative definite, sparse and ill-conditioned.

1.1.3. Semidiscretisation by Finite Elements. Instead of the finite difference discretisation from the previous section we can as well discretise (1.1) in the weak or variational form by finite elements on a uniform mesh of $[0,1]^2$ with n interior grid-points $(x_i)_{i=1}^n$, mesh width $h = (\sqrt{n}+1)^{-1}$ and piecewise linear basis functions $(\phi_i)_{i=1}^n$. The corresponding space-discrete system is (1.2), where W is defined as in (1.3), $K := K^{FEM} := E^{-1}K^{FD}$ for the matrix K^{FD} from (1.3) and $A := E^{-1}A^{FEM}$ for the matrices

$$A_{i,j}^{FEM} := \int_\Omega \langle \nabla\phi_i(\xi), \nabla\phi_j(\xi)\rangle \mathrm{d}\xi, \qquad E_{i,j} := \int_\Omega \phi_i(\xi)\phi_j(\xi)\mathrm{d}\xi. \tag{1.4}$$

The mass matrix E is symmetric positive definite, well conditioned and sparse. The system matrix $A = E^{-1}A^{FEM}$ has a negative spectrum, is non-symmetric, dense and ill-conditioned. Therefore, one avoids to work with A and instead uses a generalised formulation, see (1.7).

1.1.4. Linear State Feedback Control. The discrete optimal control u can be realised in linear state feedback form [12]

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

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

$$A^T X + XA - XKK^T X + WW^T = 0. \tag{1.5}$$

The matrix A is of size $n \times n$. The matrices KK^T and WW^T 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.

1.1.5. Solution of the Algebraic Matrix Riccati Equation. The non-linear equation (1.5) can be solved by Newton's method [11]. The initial guess $X_0 := 0$ is sufficient to guarantee global convergence, but in the context of multilevel methods a good initial guess can also be obtained by a coarser level solution in the nested iteration. In each step i of Newton's method we have to solve a Lyapunov equation

$$A_i^T X_i + X_i A_i + C_i = 0 \tag{1.6}$$

where the matrices A_i and C_i are of the form

$$A_i := A - KK^T X_{i-1}, \qquad C_i := WW^T - X_{i-1}KK^T X_{i-1}.$$

For the finite difference discretisation $A = A^{FD}$ the negative definite matrix A_i in the i-th step of Newton's method is data-sparse in the sense that only the sparse $n \times n$ matrix A, the vector K and the vector $K^T X_{i-1}$ have to be stored. For C_i we have to store W and $X_{i-1}K$ in addition.

For the finite element discretisation it is advantageous to consider the generalised Lyapunov equation:

LEMMA 1.1. *Let A, K, W denote the matrices of the space-discrete system (1.2) for the finite element discretisation (1.4). Let $\widehat{X}_i$ be the unique solution to the generalised Lyapunov equation*

$$\widehat{A}_i^T \widehat{X}_i E + E\widehat{X}_i \widehat{A}_i + \widehat{C}_i = 0, \tag{1.7}$$

where the matrices $\widehat{A}_i, \widehat{C}_i$ *are*

$$\widehat{A}_i := A^{FEM} - K^{FD}(K^{FD})^T \widehat{X}_{i-1} E, \quad \widehat{C}_i := WW^T - E\widehat{X}_{i-1} K^{FD}(K^{FD})^T \widehat{X}_{i-1} E.$$

Then the solution X_i *to (1.6) is* $X_i = E\widehat{X}_i E$.

Proof. By inserting $\widehat{X}_i = E^{-1} X_i E^{-1}$ we get

$$\begin{aligned}
\widehat{A}_i^T \widehat{X}_i E &= (A^{FEM} - K^{FD}(K^{FD})^T \widehat{X}_{i-1} E)^T E^{-1} X_i E^{-1} E \\
&= (E^{-1} A^{FEM} - E^{-1} K^{FD}(K^{FD})^T E^{-1} E \widehat{X}_{i-1} E)^T X_i \\
&= (E^{-1} A^{FEM} - K K^T X_{i-1})^T X_i \\
&= A_i^T X_i
\end{aligned}$$

and analogously for $E\widehat{X}_i \widehat{A}_i = X_i A_i$. The right-hand side fulfils

$$\widehat{C}_i = WW^T - E\widehat{X}_{i-1} K^{FD}(K^{FD})^T \widehat{X}_{i-1} E = WW^T - X_{i-1} K K^T X_{i-1} = C_i.$$

□

The matrices $\widehat{A}_i$ in the generalised Lyapunov equation (1.7) are data-sparse in the sense that A^{FEM} is sparse and the matrix $K^{FD}(K^{FD})^T \widehat{X}_{i-1} M$ of rank 1.

The Lyapunov equation (1.6) is a special Sylvester equation which is of the form

$$AX - XB + C = 0 \tag{1.8}$$

for the matrices $A := A_i^T$, $B := -A_i$ and $C := C_i$. A Sylvester equation is uniquely solvable for all matrices C if and only if the spectra of A and B are disjoint. In our setting the matrix A_i is negative definite and therefore $A < 0$ and $B > 0$ such that the existence of a unique solution is guaranteed.

In the following subsection 1.3 we determine a suitable format for an approximation to the solution X of the Sylvester equation (1.8) where the matrix C is of low rank.

1.2. Second Model Problem. The model problem of this section is identical to the linear time invariant control problem (1.1) except that the governing PDE is now

$$\dot{x}(t,\xi) = \partial_{\xi_1}^2 x(t,\xi) + \partial_{\xi_2}^2 x(t,\xi) + \beta \partial_{\xi_1} x(t,\xi) + \kappa(\xi) u(t),$$

leading to a discrete system

$$\dot{x}(t) = Ax(t) + Ku(t), \quad y(t) = W^T x(t)$$

with a non-symmetric matrix A. We aim at finding a lower order system

$$\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{K}u(t), \quad y(t) = \hat{W}^T \hat{x}(t).$$

so that $\hat{A}$ is considerably smaller than A while the input-output error is bounded and the reduced system stable [2]. The reduced system can be constructed based on a low rank approximation $\tilde{X}$ of the so-called cross Gramian X which is the solution of the Sylvester equation [2]

$$AX + XA + KW^T = 0.$$

1.3. Structure of the Solution. In the ith step of Newton's method to solve the algebraic matrix Riccati equation (1.5), we have to solve a Sylvester equation (1.8) where the matrix C is of rank at most

$$\operatorname{rank}(C) \leq \operatorname{rank}(WW^T) + \operatorname{rank}(X_{i-1}KK^TX_{i-1}) \leq 2.$$

Since the discrete system (1.2) involves a discretisation error, it is reasonable to solve the Sylvester equation only up to an accuracy ε of the size of the discretisation error, i.e., we seek an approximation $\tilde{X}$ to the solution X of (1.8) such that

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

The idea now is to choose a matrix $\tilde{X}$ that allows for a data-sparse representation.

DEFINITION 1.2. *($R(k)$-matrix representation) Let $k, n, m \in \mathbb{N}$. A matrix $R \in \mathbb{R}^{n\times m}$ is called an $R(k)$-matrix (given in $R(k)$-representation) if R is represented in factorised form*

$$R = UV^T, \qquad U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{m\times k}, \tag{1.9}$$

with U, V in full matrix representation.

The two factors in the representation (1.9) of an $R(k)$-matrix involve $k(n+m)$ values to be stored. The matrix-vector multiplication $y := Rx$ can be done in two steps involving the two matrix-vector products $z := V^Tx$ and $y := Uz$ that consist of $\mathcal{O}(k(n+m))$ basic arithmetic operations.

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 (1.9) by use of a (reduced) singular value decomposition and each matrix of the form (1.9) 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 (1.8).

THEOREM 1.3. *(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 spectrum $\sigma(A)$ and $\sigma(B)$ separated by a line (e.g., negative and positive). Then for each matrix $C \in \mathbb{R}^{n\times m}$ of rank at most k_C and each $0 < \varepsilon < 1$ there exists a matrix $\tilde{X} \in \mathbb{R}^{n\times m}$ that approximates the solution X to (1.8) by*

$$\|X - \tilde{X}\|_2 \leq \varepsilon \|X\|_2, \tag{1.10}$$

where the rank of $\tilde{X}$ is bounded by $\operatorname{rank}(\tilde{X}) \leq k_C k_\varepsilon$, $k_\varepsilon = \mathcal{O}(\log(1/\varepsilon))$.

The proof of Theorem 1.3 is given in [6] (see also [16, 1]). One should note that the rank k_ε depends on the location of the spectra of A and B. In our model problem this is $k = \mathcal{O}(\log(1/\varepsilon)\log(n))$.

1.4. Large Scale Sylvester Equations. A fixed Sylvester equation (1.8) can, e.g., be solved by the Bartels-Stewart algorithm [3], which is of complexity $\mathcal{O}(n^3)$. In the context of large scale Sylvester equations (i.e., $n > 10^5$) one is interested in reducing the complexity for a certain class of matrices A, B, C.

Hu and Reichel [20] propose to use Krylov subspace methods for the solution of the Sylvester equation. In each iterative step the equation is projected to a small dimension where one can use, e.g., the Bartels-Stewart algorithm as a solver. The authors do not exploit some kind of low rank structure but the fact that A and B allow for a fast matrix-vector multiplication. One step of their algorithm is of complexity $\mathcal{O}(nm)$ and the necessary number of iterations increases as the condition of the Sylvester equation increases.

Li and White [15] propose an iterative method for the solution of the Lyapunov equation based on the factorisation of the matrix C and the solution X. Their method is a special implementation of the classical ADI algorithm (previously proposed by Penzl [18]) and requires the solution of a shifted linear system $A - \lambda I$ in each step. The number J of steps necessary to gain a good approximation $\tilde{X}$ to X depends on the choice of the shifts λ. For the non-symmetric case there is no nontrivial upper bound for J. The main problem is that the approximation of rank J (after J steps of ADI) is not necessarily close to a best approximation of rank J.

Penzl [17] presents a multigrid method to compute the solution X to the Lyapunov equation but he does not exploit the fact that X can — at least if C is of low rank — be approximated by a low rank matrix $\tilde{X}$, therefore the complexity of one multigrid step is $\mathcal{O}(n^2)$. He also gives a convergence analysis for a simple model problem and proves that the convergence rate is bounded independently of the problem size n.

In this paper, we explain how one can compute a low rank approximation $\tilde{X}$ to the solution X of (1.8) by use of the multigrid method. We use the usual Jacobi smoother and standard prolongation and restriction operators but extend the basic multigrid cycle by a projection step $X_i \mapsto \mathcal{T}_k(X_i)$ that ensures that the rank of the ith iterate X_i is bounded. For a sufficiently large rank k the error $\|X_i - \mathcal{T}_k(X_i)\|$ due to the projection of the iterate X_i (cf. Section 2) can be regarded as the standard truncation error due to limited machine precision.

Each multigrid step is of complexity $\mathcal{O}(n+m)$ and a nested iteration combined with a level independent good convergence rate guarantees that we need only $\mathcal{O}(1)$ steps to solve the equation up to the discretisation error.

The convergence analysis for simple model problems turns out to be fairly trivial. The structure of the Sylvester equation allows us to carry results for linear systems $Ax = b$ over to the Sylvester equation such that the convergence rate can be bounded also for general domains and operators. The effect of the projection to low rank in the multigrid cycle can be regarded as a reduction of the machine precision. In our numerical tests the convergence rate is not deteriorated by the projection.

2. $R(k)$-Matrix Arithmetics. The set of $n \times m$ $R(k)$-matrices is not a linear space because the addition of two matrices of rank at most k might result in a matrix of rank larger than k. In this sense the $R(k)$-matrix format is not suitable for iterative solution schemes for the Sylvester equation.

However, $R(k)$-matrices allow for an efficient singular value decomposition such that the projection (a best approximation) to lower rank is of complexity $\mathcal{O}(k^2(n+m))$. This projection can be used to keep the iterates in the set $R(k)$.

LEMMA 2.1 (reduced SVD, truncation). *(a) Let $R = UV^T \in \mathbb{R}^{n\times m}$ be an $R(k)$-matrix. A reduced singular value decomposition of R can be computed with complexity $N_{R,\mathrm{SVD}}(n,m,k) \lesssim 6k^2(n+m) + 23k^3$ as follows:*

1. *Calculate a (reduced) QR-decomposition $U = Q_U R_U$ of U, $Q_U \in \mathbb{R}^{n\times k}, R_U \in \mathbb{R}^{k\times k}$.*
2. *Calculate a (reduced) QR-decomposition $V = Q_V R_V$ of V, $Q_V \in \mathbb{R}^{m\times k}, R_V \in \mathbb{R}^{k\times k}$.*
3. *Calculate a singular value decomposition $R_U R_V^T = \widetilde{U}\Sigma\widetilde{V}^T$.*
4. *Define $\widehat{U} := Q_U\widetilde{U}$ and $\widehat{V} := Q_V\widetilde{V}$.*

Then $R = \widehat{U}\Sigma\widehat{V}^T$ is a (reduced) SVD. Due to [5, Sections 5.2.9 and 5.4.5], the

Rank	$k=4$	$k=8$	$k=16$	$k=32$	$k=64$	$k=128$
Time	6.19	15.32	49.37	172.73	653.40	2637.5

TABLE 2.1
Time in seconds for the reduced SVD of an $n \times n$ $R(k)$-matrix, $n = 1024^2$.

complexity of the previous steps is

QR-decomposition of U:	$4nk^2$	
QR-decomposition of V:	$4mk^2$	
multiplication of $R_U R_V^T$:		$2k^3$
SVD of $R_U R_V^\top$:		$\approx 21k^3$
Multiplication of $Q_U\widetilde{U}$ and $Q_V\widetilde{V}$:	$2nk^2 + 2mk^2$	
Altogether: $N_{R,\mathrm{SVD}}(n,m,k) =$	$6k^2(n+m)+$	$23k^3$

(b) A truncation of an $R(k)$-matrix R to rank $k' \le k$ is defined as the best approximation with respect to the Frobenius and spectral norm of R in the set of $R(k')$-matrices. This can be computed by using the first k' columns of the matrices $\widehat{U}\Sigma$ and $\widehat{V}$ from the reduced singular value decomposition of R with the same complexity as above. We denote the truncation to k' by the symbol

$$\mathcal{T}_{k'}. \tag{2.1}$$

If $k' \ge k$, then $\mathcal{T}_{k'}$ is the identity. In the $R(k)$-matrix representation (1.9), the matrices U, V are extended by $k' - k$ zero columns.

We remark that the truncation in part (b) becomes non-unique when the k'-th and $(k'+1)$-st singular values are equal.

LEMMA 2.2 (spectral and Frobenius norm). *The spectral and Frobenius norm of an $n \times m$ $R(k)$-matrix R can be computed as in Lemma 2.1a with complexity $N_{R,\|\cdot\|}(n,m,k) \lesssim 4k^2(n+m) + 23k^3$.*

Proof. The norms can be obtained from the singular values, i.e., steps 1-3 from Lemma 2.1a are to be performed. □

EXAMPLE 2.3 (complexity of the truncation in practice). *We implement the truncation procedure of Lemma 2.1 on a* SUN ULTRASPARC III *with* 900 *MHz CPU clock rate and* 150 *MHz memory clock rate by use of the* LAPACK *subroutines* `dgeqrf` *and* `dgesvd` *for the QR-factorisation and singular value decomposition of full matrices. The $1024^2 \times 1024^2$ matrix R of rank k is given in $R(k)$-matrix representation and has random entries in the factors U, V. We truncate R down to rank $k/2$. The time in seconds to compute the result is given in Table 2.1.*

3. Tensor structure of the Sylvester equation. In order to formulate and analyse the iterative solvers for the Sylvester equation, we need to reformulate the matrix equation in terms of a standard linear system of equations. For notational purposes we also introduce the Kronecker product formulation.

3.1. Algebraic Structure. The Sylvester equation (1.8) can be written (for each entry (i,j)) in the form

$$\sum_{\nu=1}^{n} A_{i\nu} X_{\nu j} - \sum_{\nu=1}^{m} X_{i\nu} B_{\nu j} = -C_{ij},$$

which means that the entries of the Sylvester operator $\mathcal{S}^{A,B} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m},\ X \mapsto AX - XB$ are

$$\mathcal{S}^{A,B}_{ij,pq} = \delta_{jq}A_{ip} - \delta_{ip}B_{qj}, \qquad \delta_{jq} = \begin{cases} 1 & \text{if } j = q \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

If we order the indices columnwise (rowwise) the matrix representation is

$$\mathcal{S}^{A,B}_{col} = \begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix} - \begin{bmatrix} B_{11}I & \cdots & B_{m1}I \\ \vdots & \ddots & \vdots \\ B_{1m}I & \cdots & B_{mm}I \end{bmatrix},$$
$$\mathcal{S}^{A,B}_{row} = \begin{bmatrix} A_{11}I & \cdots & A_{1n}I \\ \vdots & \ddots & \vdots \\ A_{n1}I & \cdots & A_{nn}I \end{bmatrix} - \begin{bmatrix} B & & \\ & \ddots & \\ & & B \end{bmatrix}.$$

The Kronecker product

$$X \otimes Y := \begin{bmatrix} X_{11}Y & \cdots & X_{1n}Y \\ \vdots & \ddots & \vdots \\ X_{n1}Y & \cdots & X_{nn}Y \end{bmatrix}$$

allows us to use the short notation

$$\mathcal{S}^{A,B} := \mathcal{S}^{A,B}_{col} = I \otimes A - B^T \otimes I.$$

For the finite element discretisation it was advantageous to consider the generalised Sylvester operator

$$\mathcal{S}^{A,B,E} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m},\ X \mapsto AXE - EXB$$

which can be written in terms of the Kronecker product by

$$\mathcal{S}^{A,B,E} = E \otimes A - B^T \otimes E,$$

i.e., the entries of the matrix $\mathcal{S}^{A,B,E}$ are

$$\mathcal{S}^{A,B,E}_{ij,pq} = E_{jq}A_{ip} - E_{ip}B_{qj}.$$

3.2. Analytic Structure. In this section we want to identify the matrices $\mathcal{S}^{A,B}$ and $\mathcal{S}^{A,B,E}$ of the (generalised) Sylvester operator as the discretisation of a tensor product operator on the tensor domain $\Omega \times \Omega$. This will enable us to use proofs of multigrid convergence for the product operator.

3.2.1. Finite Element Discretisation. We consider the finite element Galerkin discretisation of the operator $\mathcal{A} : H^1_0(\Omega\times\Omega) \times H^1_0(\Omega\times\Omega) \to H^{-1}(\Omega\times\Omega)$,

$$\mathcal{A}[u](x,y) = -\sum_{\nu=1}^{2} \partial^2_{x_\nu} u(x,y) - \sum_{\nu=1}^{2} \partial^2_{y_\nu} u(x,y) \tag{3.2}$$

using the set $V_{n^2} := \{\varphi_{ij} \mid i,j = 1,\dots,n\}$ of tensor product basis functions based on the basis functions ϕ_i from Section 1.1.3:

$$\varphi_{ij}(x,y) := \phi_i(x)\phi_j(y), \qquad x,y \in \Omega.$$

The Galerkin stiffness matrix is the matrix $\mathbb{A}$ with entries

$$\begin{aligned}\mathbb{A}_{ij,pq} &= \int_\Omega\int_\Omega \langle\nabla_x\varphi_{ij}(x,y),\nabla_x\varphi_{pq}(x,y)\rangle + \langle\nabla_y\varphi_{ij}(x,y),\nabla_y\varphi_{pq}(x,y)\rangle\,\mathrm{d}x\mathrm{d}y\\ &= \int_\Omega\int_\Omega \langle\nabla\phi_i(x),\nabla\phi_p(x)\rangle\phi_j(y)\phi_q(y) + \langle\nabla\phi_j(y),\nabla\phi_q(y)\rangle\phi_i(x)\phi_p(x)\,\mathrm{d}x\mathrm{d}y\\ &= \int_\Omega \langle\nabla\phi_i(x),\nabla\phi_p(x)\rangle\,\mathrm{d}x\int_\Omega \phi_j(y)\phi_q(y)\,\mathrm{d}y +\\ &\quad + \int_\Omega \langle\nabla\phi_j(y),\nabla\phi_q(y)\rangle\,\mathrm{d}y\int_\Omega \phi_i(x)\phi_p(x)\,\mathrm{d}x\\ &= A_{ip}E_{jq} + E_{ip}A_{jq} = \mathcal{S}^{A,A,E}_{ij,pq},\end{aligned}$$

where A, E are the stiffness and mass matrices from Section 1.1.3 and $\mathcal{S}^{A,A,E}$ is the Sylvester operator from Section 3.1. Therefore, $\mathbb{A}$ is just another notation for $\mathcal{S}^{A,A,E}$, but it allows us to regard it as a standard finite element discretisation of an elliptic operator, hence standard multigrid theory can be applied.

3.2.2. Finite Difference Discretisation. For a finite difference discretisation of the operator (3.2) one can derive as in the previous section

$$\mathbb{A}^{FD}_{ij,pq} = \mathcal{S}^{A,A}_{ij,pq},\ \text{i.e.,}\quad \mathbb{A}^{FD} = \mathcal{S}^{A,A},$$

where A is the finite difference matrix from Section 1.1.2.

Before we introduce the multigrid method, we first consider one important ingredient, namely the smoother. The standard smoother used in a multigrid method is Jacobi (or Gauss-Seidel), which requires in our setting the solution of diagonal Sylvester equations, resp. diagonal generalised Sylvester equations.

4. Diagonal Sylvester Equation. A diagonal Sylvester equation

$$\begin{bmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{bmatrix} X - X \begin{bmatrix} b_1 & & \\ & \ddots & \\ & & b_m \end{bmatrix} + C = 0 \tag{4.1}$$

with $a_i < b_j$ for all $1 \le i \le n$ and $1 \le j \le m$ allows for a direct solution by

$$X_{ij} = C_{ij}/(b_j - a_i). \tag{4.2}$$

If the matrix C is of rank 1 with $R(1)$-matrix representation $C = cd^T$, then

$$X = \begin{bmatrix} c_1 & & \\ & \ddots & \\ & & c_n \end{bmatrix} \begin{bmatrix} (b_1-a_1)^{-1} & \cdots & (b_m-a_1)^{-1} \\ \vdots & \ddots & \vdots \\ (b_1-a_n)^{-1} & \cdots & (b_m-a_n)^{-1} \end{bmatrix} \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_m \end{bmatrix}.$$

The following Lemma 4.2 proves that the Cauchy matrix $\mathcal{C}_{ij} = (b_j - a_i)^{-1}$ allows for a low rank approximation (a special case of Theorem 1.3). The idea is to construct a separable representation for the function

$$f(x,y) := \frac{1}{x-y} \approx \sum_{\nu=1}^{k} g_\nu(x)h_\nu(y)$$

so that

$$\mathcal{C}_{ij} \approx \sum_{\nu=1}^{k} g_\nu(b_j) h_\nu(a_i).$$

However, if the distance between the sets

$$I_a := \{a_1, \ldots, a_n\} \quad \text{and} \quad I_b := \{b_1, \ldots, b_m\}$$

is small compared to their diameters then the separable approximation requires a large rank k. Therefore we subdivide the sets I_a and I_b into subsets $t \subset I_a$ and $s \subset I_b$ so that they fulfil the admissibility condition

$$\min\{\operatorname{diam}(t), \operatorname{diam}(s)\} \leq \operatorname{dist}(t, s). \tag{4.3}$$

An explicit construction is given in the following.

CONSTRUCTION 4.1. *(Local R(k)-matrix approximation of the Cauchy matrix) Let $t \subset I_a$ and $s \subset I_b$ fulfil (4.3) and let*

$$t_0 := \frac{1}{2}(\min_{a_i \in t} a_i + \max_{a_i \in t} a_i), \quad s_0 := \frac{1}{2}(\min_{b_j \in s} b_j + \max_{b_j \in s} b_j).$$

Then we define for $i \in t$ and $j \in s$ the approximation

$$\tilde{\mathcal{C}}_{ij} := \begin{cases} \sum_{\nu=0}^{k} (t_0 - b_j)^{-\nu-1} (t_0 - a_i)^\nu & \text{if } \operatorname{diam}(t) \leq \operatorname{diam}(s), \\ \sum_{\nu=0}^{k} (a_i - s_0)^{-\nu-1} (b_j - s_0)^\nu & \text{otherwise.} \end{cases}$$

The matrix $\tilde{\mathcal{C}}_{i \in t, j \in s}$ is an R(k)-matrix where the factors U, V are

$$U_{i\nu} := \begin{cases} (t_0 - a_i)^\nu & \text{if } \operatorname{diam}(t) \leq \operatorname{diam}(s), \\ (a_i - s_0)^{-\nu-1} & \text{otherwise,} \end{cases}$$

$$V_{j\nu} := \begin{cases} (t_0 - b_j)^{-\nu-1} & \text{if } \operatorname{diam}(t) \leq \operatorname{diam}(s), \\ (b_j - s_0)^\nu & \text{otherwise.} \end{cases}$$

LEMMA 4.2. *(Local approximation error) Let t, s and $\tilde{\mathcal{C}}$ be as in Construction 4.1 and let $0 < \varepsilon < 1$. Then*

$$|\tilde{\mathcal{C}}_{ij} - \mathcal{C}_{ij}| \leq \varepsilon |\mathcal{C}_{ij}| \tag{4.4}$$

holds for all $i \in t$,$j \in s$ and a rank

$$k := \lceil \log_3(1/\varepsilon) \rceil + 1.$$

Proof. Without loss of generality we assume $\operatorname{diam}(t) \leq \operatorname{diam}(s)$. The exact Taylor expansion of f with respect to x is

$$f(x, y) = \sum_{\nu=0}^{\infty} \frac{1}{\nu!} \partial_x^\nu f(t_0, b_j)(a_i - t_0)^\nu = \sum_{\nu=0}^{\infty} (t_0 - b_j)^{-\nu-1} (t_0 - a_i)^\nu.$$

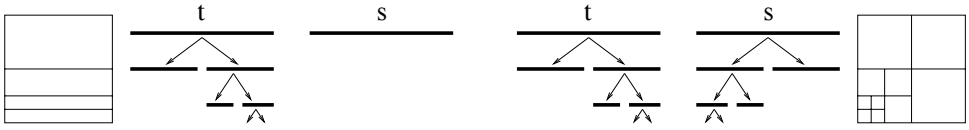

FIG. 4.1. *Recursive subdivision of one (left) or two (right) subintervals and the corresponding partitions of the Cauchy matrix* $\mathcal{C}$.

Using this representation and the assumption (4.3) we get

$$\begin{aligned}
|\tilde{\mathcal{C}}_{ij} - \mathcal{C}_{ij}| &= \Big|\sum_{\nu=k}^{\infty}(t_0-b_j)^{-\nu-1}(t_0-a_i)^{\nu}\Big| \leq \sum_{\nu=k}^{\infty}|t_0-b_j|^{-\nu-1}|t_0-a_i|^{\nu} \\
&\leq \sum_{\nu=k}^{\infty}\left(\operatorname{dist}(t,s)+\frac{1}{2}\operatorname{diam}(t)\right)^{-\nu}\left(\frac{1}{2}\operatorname{diam}(t)\right)^{\nu}|t_0-b_j|^{-1} \\
&\overset{(4.3)}{\leq} \sum_{\nu=k}^{\infty}3^{-\nu}|t_0-b_j|^{-1} = 3^{-k+1}\frac{1}{2}|t_0-b_j|^{-1} \leq \varepsilon|\mathcal{C}_{ij}|.
\end{aligned}$$

□

In order to satisfy the admissibility condition (4.3) there are two strategies:

First, we can subdivide the set t recursively into two parts t_1 and t_2 of half the diameter so that one of the two is admissible to s (cf. Figure 4.1). The other one is then further subdivided until the diameter is less than the distance to s. This strategy produces blocks $t' \times s$, $t' \subset t$, for which we can apply Construction 4.1. The number of blocks is $p := \lceil \log_2(\frac{\operatorname{diam}(t)}{\operatorname{dist}(t,s)})\rceil + 1$. In total we have to store and compute $\mathcal{O}(pk(n+m))$ entries of the R(k)-matrix representation.

Second, we can subdivide always both sets s and t each into two parts of half the diameter so that three of the four pairs are admissible and the fourth one has to be subdivided further (cf. Figure 4.1). This strategy will then produce $p := 3\lceil \log_2(\frac{\operatorname{diam}(t)}{\operatorname{dist}(t,s)})\rceil + 1$ blocks (more than the first strategy), but they are of different size which is decaying geometrically. Therefore we have to store and compute only $\mathcal{O}(k(n+m))$ entries of the R(k)-matrix representations.

In both cases, the rank of the approximation $\tilde{\mathcal{C}}$ is pk. In the second case we can exploit the hierarchical structure for the efficient computation of an approximation for X. We will give the details later in Construction 4.5.

COROLLARY 4.3 (Approximation error). *Let* $\tilde{\mathcal{C}}$ *be an approximation to the Cauchy matrix with relative error* ε*, i.e.,* $|\tilde{\mathcal{C}}_{ij} - \mathcal{C}_{ij}| \leq \varepsilon|\mathcal{C}_{ij}|$ *for all* $1 \leq i \leq n$ *and* $1 \leq j \leq m$*. Let* C *be an* $R(k_C)$*-matrix with entries* $C_{ij} = \sum_{\nu=1}^{k_C} c_i^{(\nu)} d_j^{(\nu)}$*. Then the matrix* $\tilde{X}_{ij} := \sum_{\nu=1}^{k_C} c_i^{(\nu)} \tilde{\mathcal{C}}_{ij} d_j^{(\nu)}$ *approximates the solution* X *to (4.1) by*

$$|X_{ij} - \tilde{X}_{ij}| \leq \varepsilon|X_{ij}|, \qquad \|X - \tilde{X}\|_F \leq \varepsilon\|X\|_F.$$

Proof.

$$|X_{ij} - \tilde{X}_{ij}| = |\sum_{\nu=1}^{k_C} c_i^{(\nu)} d_j^{(\nu)}|\,|\tilde{\mathcal{C}}_{ij} - \mathcal{C}_{ij}| \leq \varepsilon|\sum_{\nu=1}^{k_C} c_i^{(\nu)} d_j^{(\nu)}|\,|\mathcal{C}_{ij}| = \varepsilon|X_{ij}|.$$

□

REMARK 4.4 (adaptive choice of the rank). *In practice, one is interested in a good approximation $\tilde{\mathcal{C}}$ to the Cauchy matrix $\mathcal{C}$, preferably an approximation with minimal rank for a prescribed accuracy ε. Our construction only yields a suboptimal candidate where the rank is higher than necessary. In the multigrid method the approximation $\tilde{\mathcal{C}}$ will be used several times, such that it pays to spend more effort in the computation of $\tilde{\mathcal{C}}$. One way to do this is to compute a candidate $\tilde{\mathcal{C}}_1$ as above up to accuracy $\varepsilon/10$ and compute an approximant $\tilde{\mathcal{C}}$ to $\tilde{\mathcal{C}}_1$ up to accuracy ε with minimal rank by use of the reduced singular value decomposition of Lemma 2.1.*

The construction of a good *low rank* approximant $\tilde{\mathcal{C}}$ to the Cauchy matrix bears two bottlenecks:

First, one has to store a matrix of rank pk. For a large scale problem with $n = m = 10^6$, $|b_1 - a_n| = 10^{-3}$, $|a_n - a_1| = 1$ and $\varepsilon = 10^{-6}$ there are more than 300 million entries to be stored which requires more than two Gigabyte of memory in double precision arithmetic.

Second, the estimated rank pk (in the above example $pk = 154$) is typically too large. The truncation to lower rank is of quadratic complexity in the rank which means prohibitively expensive (cf. Example 2.3).

CONSTRUCTION 4.5 (hierarchical construction). *Assume that $\mathcal{C}$ is subdivided as indicated in Figure 4.1. The construction consists of three parts. In part one we define the blockwise local approximation of $\mathcal{C}$. In part two we construct a blockwise local approximation of X and in part three we combine the blocks to a global approximation $\tilde{X}$ of X.*

Part 1. *For each of the blocks $t \times s \subset I_a \times I_b$ we define the approximation $\tilde{\mathcal{C}}|_{t\times s}$ as in Construction 4.1 but with a different target accuracy by taking the rank $k' := \lceil \log_3(\varepsilon^{-1}/3) \rceil + 1$.*

Part 2. *Let $C = \sum_{\nu=1}^{k_C} c^{(\nu)}(d^{(\nu)})^T$. The matrix X is blockwise of the form*

$$X|_{t\times s} = \sum_{\nu=1}^{k_C} \operatorname{diag}(c^{(\nu)}|_t)\, \mathcal{C}|_{t\times s} \operatorname{diag}(d^{(\nu)}|_s)$$

and we approximate it by

$$\tilde{X}'|_{t\times s} := \sum_{\nu=1}^{k_C} \operatorname{diag}(c^{(\nu)}|_t)\, \tilde{\mathcal{C}}|_{t\times s} \operatorname{diag}(d^{(\nu)}|_s).$$

At last we recompress the matrix blockwise by use of the reduced SVD to find a minimal rank approximation $\tilde{X}''|_{t\times s}$ so that

$$\|\tilde{X}'|_{t\times s} - \tilde{X}''|_{t\times s}\|_F \le \frac{\varepsilon}{3}\|\tilde{X}'|_{t\times s}\|_F.$$

Part 3. *We define $\tilde{X}$ recursively, starting with the largest block $I_a \times I_b$ and going down recursively. On each level (level number $\ell = 1$ for the largest block, level number $\ell = 2$ for the blocks of half the size and so forth) with corresponding block $t \times s$ we prescribe an accuracy of*

$$\varepsilon_\ell := 2^{-\ell}\frac{\varepsilon}{3}\|\tilde{X}''\|_F.$$

Let $t \times s$ be a block that is subdivided into $t_1 \times s_1, t_2 \times s_1, t_1 \times s_2, t_2 \times s_2$. Then we define

$$\tilde{X}^{t,s} := \mathcal{T}_k\left(\left[\begin{array}{c|c} \tilde{X}''|_{t_1\times s_1} & \tilde{X}''|_{t_1\times s_2} \\ \hline \tilde{X}^{t_2,s_1} & \tilde{X}''|_{t_2\times s_2} \end{array}\right]\right)$$

by use of the truncation operator from (2.1) and a target accuracy ε_ℓ in the absolute Frobenius norm.

The relative Frobenius norm accuracy $\|X - \tilde{X}\|_F / \|X\|_F$ for the approximation $\tilde{X} := \tilde{X}^{I_a, I_b}$ will be estimated in the following Lemma.

LEMMA 4.6. *For the approximation $\tilde{X}$ from Construction 4.5 holds*

$$\|X - \tilde{X}\|_F \leq \varepsilon \|X\|_F + \mathcal{O}(\varepsilon^2).$$

The complexity of the hierarchical construction is $\mathcal{O}((n+m)(k_C^2(k')^2 + k_{final}^2))$, where k' is the rank used for the local approximation of the Cauchy matrix, k_C is the rank of the matrix C and k_{final} is the rank used for the approximation of the solution X.

Proof. 1) Approximation error.
The Cauchy matrix approximation in part 1 of Construction 4.5 was chosen such that $|\mathcal{C}_{ij} - \tilde{\mathcal{C}}_{ij}| \leq \varepsilon |\mathcal{C}_{ij}|/3$. From Corollary 4.3 we conclude $\|\tilde{X}' - X\|_F \leq \varepsilon \|X\|_F/3$. In part 2 of Construction 4.5 the matrix $\tilde{X}'$ is recompressed so that

$$\|X - \tilde{X}''\|_F \leq \|X - \tilde{X}'\|_F + \|\tilde{X}' - \tilde{X}''\|_F \leq \varepsilon \|X\|_F/3 + \varepsilon \|X\|_F/3 + \mathcal{O}(\varepsilon^2).$$

Next we will show that $\|\tilde{X}'' - \tilde{X}\|_F \leq \varepsilon \|X\|_F/3 + \mathcal{O}(\varepsilon^2)$ which gives the desired estimate.

The truncation accuracy in part 3 of Construction 4.5 yields on each level ℓ of a block $t \times s$

$$\|\tilde{X}^{t,s} - \tilde{X}''|_{t\times s}\|_F \leq \varepsilon_\ell = 2^{-\ell} \varepsilon \|\tilde{X}''\|_F/3.$$

Over all levels $\ell = 1, \ldots$ this sums up to

$$\sum_{\ell=1}^{\infty} \varepsilon_\ell = \frac{1}{3}\varepsilon \|\tilde{X}''\|_F = \frac{1}{3}\varepsilon \|X\|_F + \mathcal{O}(\varepsilon^2).$$

2) Complexity.
Part 1 of Construction 4.5 is of complexity $2^{1-\ell} k' n$ for a block on level ℓ. On each level there are at most three blocks so that this sums up to

$$\sum_{\ell=1}^{\infty} 2^{1-\ell} 3k'n \leq 6k'n.$$

In part 2 we truncate each of the blocks (we neglect the diagonal scaling). Due to Lemma 2.1 the complexity is bounded by

$$\sum_{\ell=1}^{\infty} 2^{1-\ell} 3n(k_C k')^2 \leq 6k_C^2 (k')^2 n.$$

At last we combine the blocks levelwise. On each level we add four matrices, each of rank at most k_{final}, so that the complexity is bounded by

$$\sum_{\ell=1}^{\infty} 2^{1-\ell} n k_{final}^2 \leq 2k_{final}^2 n.$$

□

In order to illustrate the benefits and the complexity of Construction 4.1 and the alternatives from Remark 4.4 and Construction 4.5 we test the method for a simple artificial model problem.

EXAMPLE 4.7. *The entries of the diagonal matrices A and B are*

$$a_i = -i, \qquad b_j = j, \qquad 1 \le i, j \le 1024^2.$$

We want to approximate the solution X and the Cauchy matrix $\mathcal{C}$ for a matrix C of rank $k_C = 5$ up to an accuracy of $\varepsilon := 10^{-6}$ by approximations $\tilde{\mathcal{C}}$ and $\tilde{X}$ of minimal rank.

According to Construction 4.5 we compute the approximant in three steps:

1. *(Part 1) Hierarchical Approximation of $\mathcal{C}$ by $\tilde{\mathcal{C}}$. Since the entries of $\tilde{\mathcal{C}}$ are derived analytically, this is very fast. The blockwise rank k is 15 (as defined in Construction 4.5).*
2. *(Part 2) Blockwise approximation in the $R(k)$-matrix format.*
3. *(Part 3) Hierarchical conversion to the $R(k)$-matrix format.*

The following table displays the times the three steps take and the amount of storage needed (in the first step for $\tilde{\mathcal{C}}$ and in the second and third step for $\tilde{X}''$ and $\tilde{X}$ respectively). The numerical tests were performed on a SUN UltraSPARC III with 900 MHz CPU clock rate and 150 MHz memory clock rate.

	time (seconds)	*storage (Megabyte)*
Part 1	10.3	720
Part 2	943	180
Part 3	422	360

The amount of storage needed in Step 1 can be omitted by immediate truncation of each block to lower rank. The final approximation $\tilde{X}$ has a rank of $k_{\tilde{X}} = 22$.

The previous example illustrates that the hierarchical truncation is an efficient way to generate either a best approximation to the Cauchy matrix or to the solution of a diagonal Sylvester equation. In practice, we will use the construction to solve diagonal Sylvester equations as they appear in the Jacobi iteration. There, the diagonal entries are of similar size, i.e., the matrix is well conditioned such that the number of levels is small (typically one). The situation simplifies if all diagonal entries are equal:

EXAMPLE 4.8. *The entries of the diagonal matrices A and B are*

$$a_i \equiv a, \qquad b_j \equiv b, \qquad 1 \le i \le n, 1 \le j \le m.$$

Then the Cauchy matrix is $\mathcal{C} = (b-a)^{-1} I$ (I is the identity) and the solution is $X = (b-a)^{-1} C$.

In the following example we want to compare our construction with an iterative scheme that approximates the solution X to (4.1). For this example we fix a matrix C of rank $k_C := 5$.

EXAMPLE 4.9. *The entries of the diagonal matrices A and B are*

$$a_i = -i, \qquad b_j = j, \qquad 1 \le i, j \le 1024^2.$$

The ADI iteration from [18] to solve $AX - XB + C = 0$ starts with $X_0 := 0$ and generates the matrices

$$\begin{aligned} X_{i+1} := (A - p_i I)(A + p_i I)^{-1}\, X_i\, (A - p_i I)(A + p_i I)^{-1} \\ -2p_i (A + p_i I)^{-1}\, C\, (A + p_i I)^{-1}, \end{aligned}$$

where the parameters p_i for J steps of the iteration are given by $\kappa := (a_1/a_n)^{1/J}, t_0 := a_1, t_j := \kappa\, t_{j-1}, p_j := -\sqrt{t_{j-1}t_j}$ for $j = 1, \dots, J$. This parameter choice allows for an explicit bound on the relative error, such that the number J of steps can be determined a priori. The rank of the resulting approximant $\tilde{X}^{ADI}$ to X is equal to Jk_C. In this example the number of iterations $J = 81$ ensures that the a priori error bound is less than 10^{-6}.

The computation of an approximation $\tilde{X}^{ADI}$ takes ca. 580 seconds (this time can be reduced by using the ADI variant from [15]). However, the rank used in the representation of $\tilde{X}^{ADI}$ is $k = 405$ so that a truncation to smaller (minimal) rank would require approximately 25000 seconds (cf. Table 2.1). Alternatively, one could truncate in intermediate steps (no control of the accuracy) so that the time reduces to approximately 5000 seconds. The complexity is higher than for the hierarchical Construction 4.5 because the local blockwise ranks are much smaller than the global rank Jk_C from the ADI iteration.

4.1. Diagonal generalised Sylvester equation. At last we want to comment on diagonal *generalised* Sylvester equations. There, the system

$$\begin{bmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{bmatrix} X \begin{bmatrix} e_1 & & \\ & \ddots & \\ & & e_m \end{bmatrix} - \begin{bmatrix} \hat{e}_1 & & \\ & \ddots & \\ & & \hat{e}_n \end{bmatrix} X \begin{bmatrix} b_1 & & \\ & \ddots & \\ & & b_m \end{bmatrix} + C = 0$$

has to be solved. We assume that $e_j > 0, \hat{e}_i > 0, a_i < 0$ and $b_j > 0$ for all entries of the diagonal matrices. This system can by multiplication with $\mathrm{diag}(\hat{e}_1^{-1}, \dots, \hat{e}_n^{-1})$ from the left and $\mathrm{diag}(e_1^{-1}, \dots, e_m^{-1})$ from the right be transformed into a standard Sylvester equation for which the techniques from above are applicable, in particular

$$X_{ij} = \hat{e}_i^{-1} C_{ij} e_j^{-1} / (b_j/e_j - a_i/\hat{e}_i). \tag{4.5}$$

5. Smoothing Iterations. In this section we will consider possible smoothing iterations that are useful in the context of the multigrid method. The two simplest ones are Richardson and Jacobi, and these will be given in detail in the following.

5.1. Richardson Iteration. For linear systems of equations $Mx = b$ the (stationary) Richardson iteration is defined by

$$x_0 := 0, \qquad x_i := x_{i-1} - \theta(Mx_{i-1} - b) \qquad \text{for } i \geq 1.$$

Convergence is guaranteed for positive definite matrices M if the parameter $\theta \in \mathbb{R}$ fulfils $0 < \theta < 2\|M\|_2^{-1}$ (see, e.g., [8] and also for a generalisation to non-symmetric systems). For the linear system $AX - XB + C = 0$ the iteration reads

$$X_0 := 0, \qquad X_i := X_{i-1} - \theta(AX_{i-1} - X_{i-1}B + C) \qquad \text{for } i \geq 1. \tag{5.1}$$

The optimal damping factor is $\theta = 2/(\|M\|_2 + \|M^{-1}\|_2^{-1})$ which can be estimated by $\theta \approx \frac{3}{2}\|M\|_2^{-1}$, where $\|M\|_2 = \|A\|_2 + \|B\|_2$ is easily computable via the power iteration (here we assumed $A < 0$ and $B > 0$).

If the iterate X_{i-1} is an $R(k)$-matrix and the right-hand side C is an $R(k_C)$-matrix, then the next iterate X_i is an $R(2k + k_C)$-matrix whose representation can be computed by k matrix-vector multiplications for the matrices A and B. In order to stay in the set of $R(k)$-matrices one can truncate the resulting matrix X_i to lower rank k. This will be called the $R(k)$-Richardson iteration:

$$X_0 := 0, \qquad X_i := \mathcal{T}_k(X_{i-1} - \theta(AX_{i-1} - X_{i-1}B + C)) \qquad \text{for } i \geq 1. \tag{5.2}$$

LEMMA 5.1. *Let $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ be data-sparse matrices in the sense that the matrix-vector multiplication for A and B can be performed with complexity $\mathcal{O}(n)$ and $\mathcal{O}(m)$.*
(a) One step of the Richardson iteration (5.1) is of complexity $\mathcal{O}(nm)$.
(b) One step of the $R(k)$-Richardson iteration (5.2) is of complexity $\mathcal{O}(k^2(n+m))$.

Although the Richardson iteration is convergent for sufficiently small θ, the rate of convergence can be poor. In the context of multigrid methods one is not necessarily interested in convergence properties but in the smoothing property (cf. [8]). The next Lemma provides the necessary assumptions.

LEMMA 5.2. *Let $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ be symmetric with spectra on two disjoint halfplanes: $\sigma(A) > \sigma(B)$. Let (the mass matrix) E be symmetric positive definite. Then the Sylvester operators $\mathcal{S}^{A,B}$ and $\mathcal{S}^{A,B,E}$ are both symmetric positive definite.*

Proof. If $\sigma(A) = \{\lambda_1, \dots, \lambda_n\}$ and $\sigma(B) = \{\mu_1, \dots, \mu_m\}$, then the nm eigenvalues of the (linear) Sylvester operator $S : X \mapsto AX - XB$ are $\lambda_i - \mu_j$. By assumption all eigenvalues are positive. The symmetry follows from $\mathcal{S}^{A,B}_{ij,pq} = \delta_{jq}A_{ip} - \delta_{ip}B_{qj} = \delta_{qj}A_{pi} - \delta_{pi}B_{jq} = \mathcal{S}^{A,B}_{pq,ij}$. Analogously symmetry holds for the generalised Sylvester operator $\mathcal{S}^{A,B,E}$. From $\sigma(A) > \sigma(B)$ and the symmetry of A, B we conclude $\sigma(E^{-\frac{1}{2}}AE^{-\frac{1}{2}}) > \sigma(E^{-\frac{1}{2}}BE^{-\frac{1}{2}})$. From the first part we know $S = I \otimes E^{-\frac{1}{2}}AE^{-\frac{1}{2}} - E^{-\frac{1}{2}}BE^{-\frac{1}{2}} \otimes I > 0$ and thus by multiplying $E^{\frac{1}{2}} \otimes E^{\frac{1}{2}}$ from the left and right: $\mathcal{S}^{A,B,E} = E \otimes A - B \otimes E > 0$. ∎

5.2. Jacobi Iteration. The *Jacobi iteration* is defined by

$$X_0 := 0, \qquad X_i := X_{i-1} - \theta \mathrm{diag}(\mathcal{S})^{-1}(AX_{i-1} - X_{i-1}B + C) \qquad \text{for } i \geq 1, \quad (5.3)$$

where $\mathcal{S}$ is the Sylvester operator. The diagonal entries of the Sylvester operator are

$$\mathcal{S}^{A,B}_{ij,ij} = A_{ii} - B_{jj}, \quad \mathcal{S}^{A,B,E}_{ij,ij} = E_{jj}A_{ii} - E_{ii}B_{jj},$$

so that the corresponding Sylvester equations are

$$\mathrm{diag}(A)X - X\mathrm{diag}(B) = C_i, \quad \mathrm{diag}(A)X\mathrm{diag}(E) - \mathrm{diag}(E)X\mathrm{diag}(B) = \tilde{C}_i.$$

We have to solve the diagonal (generalised) Sylvester equations for the right-hand side $C_i = AX_{i-1} - X_{i-1}B + C$ and $\tilde{C}_i = AX_{i-1}E - EX_{i-1}B + C$ respectively. The solution is given by (4.2) and (4.5). The optimal damping factor for the Jacobi iteration is $\theta := 2/(\Lambda + \lambda)$ [8], where Λ and λ are the best bounds for

$$\lambda \mathrm{diag}(M) \leq M \leq \Lambda \mathrm{diag}(M), \qquad M = \mathcal{S}^{A,B} \text{ or } \mathcal{S}^{A,B,E}.$$

Later we will use the parameter $\theta := 1/2$ which is sufficient to guarantee the smoothing property [8] needed for the multigrid method.

If the iterate X_{i-1} is an $R(k)$-matrix and the right-hand side C is an $R(k_C)$-matrix, then the right-hand side is an $R(2k + k_C)$-matrix. A low rank approximation X_{i+1} to the solution of the diagonal Sylvester equation can be computed by means of the hierarchical Construction 4.5. The effect is the same if we solve the diagonal equation exactly and truncate the result to a fixed rank k or a fixed accuracy ε. Therefore, the $R(k)$-Jacobi iteration can be written in the form

$$X_0 := 0, \qquad X_i := \mathcal{T}_k(X_{i-1} - \theta \mathrm{diag}(\mathcal{S})^{-1}(AX_{i-1} - X_{i-1}B + C)). \qquad (5.4)$$

LEMMA 5.3. *Let $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ be data-sparse matrices in the sense that the matrix-vector multiplication for A and B can be performed with complexity $\mathcal{O}(n)$ and $\mathcal{O}(m)$.*
(a) One step of the Jacobi iteration (5.3) is of complexity $\mathcal{O}(nm)$.
(b) One step of the $R(k)$-Jacobi iteration (5.4) is of complexity $\mathcal{O}(k^2(n+m))$.

5.3. ADI Iteration as a Smoother. Apart from Richardson and Jacobi there are many other popular smoothers like Gauss-Seidel, SOR, ILU etc.. Since these are not compatible with the low rank format, they are not of interest here. The only notable exception that we are aware of is the ADI iteration from Example 4.9. There, we have to solve systems of the form

$$Ax = b$$

which can be accomplished, e.g. by a multigrid method. However, one has to be careful with the choice of the shift parameters p_i, since the optimal parameters for the smoothing property differ from the usual ones that yield the optimal convergence rate [7].

For sure, the Richardson iteration is the most simple of the smoothers under consideration. The Jacobi iteration is necessary for non-uniform grids (e.g., locally refined) in order to get mesh-independent good convergence rates. The same goal is reached by the ADI iteration.

6. Multigrid Method. The Richardson and Jacobi iteration introduced in the previous section smooth the defect in the multigrid method on one level (=grid). In the multigrid method we transfer the smoothed defect to a coarser grid and compute a defect correction on the coarser grid. The coarse grid correction is then transferred to the fine grid in order to reduce the smooth parts of the defect. On the coarsest level we use a standard solver for the Sylvester equation. For the transfer between different grids ranging from coarse ($n_0 = 9$ degrees of freedom) to fine ($n_8 = 1046529$ degrees of freedom) we need the prolongation and restriction operator defined in the following. Whereas the Richardson and Jacobi iteration had to be adopted to the low rank setting, this is not necessary for the grid transfer operators.

Let $X \in \mathbb{R}^{n\times n}$ be a matrix, and let $\hat{p} : \mathbb{R}^n \to \mathbb{R}^m$ be a linear mapping, the so-called prolongation. Then the corresponding matrix mapping p is defined by

$$p(X) := \hat{p} X \hat{p}^T. \tag{6.1}$$

The adjoint operator r, the so-called restriction, is given by

$$r(Y) := \hat{p}^T Y \hat{p} \tag{6.2}$$

for $Y \in \mathbb{R}^{m\times m}$. Since the linear mapping p does not increase the rank of a matrix, we stay in the set of $R(k)$ matrices (only of different size n, m). Moreover, if $X = AB^T$ is an $R(k)$-matrix, then

$$p(X) = (\hat{p}A)(\hat{p}B)^T,$$

so that the prolonged (or restricted in case $r(Y)$) matrix is naturally given in the desired $R(k)$-format. In the notation of Section 3.1 the prolongation is of the tensor structure $p = \hat{p} \otimes \hat{p}$.

6.1. Multigrid Algorithm and Convergence Results. Let $\ell = 0, \dots, L$ be the level numbers and assume that on each level we have a discrete linear equation[1]

$$\mathfrak{A}_\ell \mathfrak{x}_\ell = \mathfrak{b}_\ell \qquad (0 \le \ell \le L)$$

with symmetric and positive definite $n_\ell \times n_\ell$ matrices $\mathfrak{A}_\ell$, while $\mathfrak{b}_\ell$ is some right-hand side and $\mathfrak{x}_\ell$ the corresponding solution. For some domains (e.g., the unit square from our model problem) the hierarchy $(\mathfrak{A}_\ell)_{\ell=0}^L$ of discrete problems is naturally given by successive refinement of the coarsest grid. For more complicate domains one needs suitable coarsening algorithms, e.g., composite finite elements [10] or algebraic multigrid [19, 21].

We recall the general multigrid algorithm (for details of the algorithm or the following statements we refer to Hackbusch [7], [8]):

function $MGM(\ell, \mathfrak{x}, \mathfrak{b})$; (returns the new iterate)
if $\ell = 0$ **then** $\mathfrak{x} := \mathfrak{A}_0^{-1} f$ **else**
begin
  for $i := 1$ **to** ν **do** $\mathfrak{x} := \mathcal{S}_\ell(\mathfrak{x}, \mathfrak{b})$; (presmoothing)
  $\mathfrak{d} := r(\mathfrak{A}_\ell \mathfrak{x} - \mathfrak{b})$; (restriction of the defect)
  $\mathfrak{y} := 0$; (starting value for the corrections)
  for $i := 1$ **to** γ **do** $v := MGM(\ell - 1, \mathfrak{y}, \mathfrak{d})$;
  $\mathfrak{x} = \mathfrak{x} - p\mathfrak{y}$; (coarse-grid correction)
  for $i := 1$ **to** ν **do** $\mathfrak{x} := \mathcal{S}_\ell(\mathfrak{x}, \mathfrak{b})$; (postsmoothing)
end;
$MGM := \mathfrak{x}$ (new iterate returned)

The V-cycle (W-cycle) corresponds to $\gamma = 1$ ($\gamma = 2$). ν is the number of pre- and postsmoothing steps using the smoothing procedure $\mathcal{S}_\ell$ (e.g., Richardson, Jacobi or the $R(k)$-counterparts). p is the prolongation from (6.1), e.g., the piecewise linear interpolation in the case of difference schemes, or the canonical finite element transfer in the case of finite element subspaces $V_{\ell-1} \subset V_\ell$).

The essential conditions for the convergence of the W-cycle are the smoothing and approximation properties. A simplified version of the smoothing property is

$$\|\mathfrak{A}_\ell \mathfrak{S}_\ell^\nu\| \le C_{\mathrm{sm}} \|\mathfrak{A}_\ell\| \, \sigma_\ell \eta(\nu) \qquad \text{for } \nu \ge 1 \text{ with } \lim_{\nu \to \infty} \eta(\nu) = 0, \tag{6.3}$$

where $\mathfrak{S}_\ell$ is the iteration matrix of the iteration $\mathcal{S}_\ell$ (i.e., $\mathcal{S}_\ell(\mathfrak{x}, \mathfrak{b}) = \mathfrak{S}_\ell \mathfrak{x} + \mathfrak{T}_\ell \mathfrak{b}$) and C_{sm} is a constant independent of ℓ, while σ_ℓ is any scaling quantity (except of Section 6.6, only $\sigma_\ell = 1$ will occur). For convenience, $\|\cdot\|$ may be considered as spectral norm, but other norms are possible. Often $\eta(\nu)$ equals

$$\eta_0(\nu) := \nu^\nu / (\nu + 1)^{\nu+1} \tag{6.4}$$

The approximation property reads

$$\|\mathfrak{A}_\ell^{-1} - p\mathfrak{A}_{\ell-1}^{-1} r\| \le C_{\mathrm{app}} / (\|\mathfrak{A}_\ell\| \, \sigma_\ell) \tag{6.5}$$

with an ℓ-independent constant C_{a} and the same scaling quantity σ_ℓ as in (6.3). Under these assumptions (and simple technical conditions on p, r and $\mathfrak{S}_\ell$), the W-cycle converges with the rate $const \cdot \eta(\nu)$ (under standard symmetry conditions on p, r and $\mathfrak{S}_\ell$, even $\nu = 1$ leads to convergence).

[1] The fracture style letters indicate matrices and vectors which will be later identified with corresponding quantities of the Sylvester equation. For instance, the vector $\mathfrak{x}_\ell$ will become the unknown solution matrix X_ℓ.

6.2. Approximation Property. Assuming a finite element discretisation with subspaces $V_{\ell-1} \subset V_\ell$ with quasi-uniform grid sizes $h_{\ell-1}$, h_ℓ ($h_{\ell-1}/h_\ell \leq const$), one obtains the estimate $\|\mathfrak{A}_\ell^{-1} - p\mathfrak{A}_{\ell-1}^{-1} r\| \leq const \cdot \|\mathfrak{A}_\ell^{-1}\| h_\ell^2$ for the spectral norm, provided that full regularity holds, i.e., the underlying boundary value problem satisfies $\|u\|_{H^2(\omega)} \leq const \, \|f\|_{L^2(\omega)}$ for the solution of $Lu = f$. If the coefficients are sufficiently smooth and ω is convex (or image of a convex domain under a smooth mapping), full regularity holds (cf. Hackbusch [9, Thm 9.1.22]). In our application, the domain ω is the product $\Omega \times \Omega$. Convexity of Ω implies convexity of ω.

Since the scaling of the stiffness matrix is such that $\|\mathfrak{A}_\ell\| \|\mathfrak{A}_\ell^{-1}\|$ is proportional to h_ℓ^{-2}, the inequality $\|\mathfrak{A}_\ell^{-1} - p\mathfrak{A}_{\ell-1}^{-1} r\| \leq const \cdot \|\mathfrak{A}_\ell^{-1}\| h_\ell^2$ is equivalent to (6.5) with $\sigma_\ell := 1$.

Weaker regularity can also be treated (see Section 6.6).

6.3. Smoothing Property without Truncation. First we consider the Richardson iteration

$$\mathcal{S}_\ell(\mathfrak{x}_\ell, \mathfrak{b}_\ell) = \mathfrak{S}_\ell \mathfrak{x}_\ell + \mathfrak{T}_\ell \mathfrak{b}_\ell \qquad \text{with } \mathfrak{S}_\ell = I - \vartheta_\ell \mathfrak{A}_\ell, \quad \mathfrak{T}_\ell \mathfrak{b}_\ell = \vartheta_\ell \mathfrak{b}_\ell$$

with the damping factor $\vartheta_\ell = 1/\|\mathfrak{A}_\ell\|_2$ (also $\vartheta_\ell = 1/\rho(\mathfrak{A}_\ell)$ because of the symmetry of $\mathfrak{A}_\ell$).

LEMMA 6.1 ([8, Thm 10.6.5]). *Let $\mathfrak{A}_\ell$ be symmetric and positive definite. Then the Richardson iteration with $\vartheta_\ell = 1/\|\mathfrak{A}_\ell\|_2$ satisfies the smoothing property with $\eta(\nu) = \eta_0(\nu)$ from (6.4) and $C_{\mathrm{sm}} \sigma_\ell = 1$.*

For the application to the Sylvester equation, we have to make use of $\vartheta_\ell = 1/\rho(\mathfrak{A}_\ell) = 1/(\max_{\lambda \in \sigma(A)} \lambda - \min_{\mu \in \sigma(B)} \mu)$. Since the matrices $\mathfrak{A}_\ell = \mathcal{S}^{A,B}$ and $\mathfrak{A}_\ell = \mathcal{S}^{A,B,M}$ are symmetric and positive definite (Lemma 5.2), the previous Lemma applies.

Next, we consider the damped Jacobi iteration

$$\mathcal{S}_\ell(\mathfrak{x}_\ell, \mathfrak{b}_\ell) = \mathfrak{S}_\ell \mathfrak{x}_\ell + \mathfrak{T}_\ell \mathfrak{b}_\ell \qquad \text{with } \mathfrak{S}_\ell = I - \vartheta_\ell \mathfrak{D}_\ell^{-1} \mathfrak{A}_\ell, \quad \mathfrak{T}_\ell \mathfrak{b}_\ell = \vartheta_\ell \mathfrak{D}_\ell^{-1} \mathfrak{b}_\ell, \tag{6.6}$$

where $\mathfrak{D}_\ell$ is the diagonal part of $\mathfrak{A}_\ell$ with ϑ_ℓ such that $\vartheta_\ell \rho(\mathfrak{D}_\ell^{-1} \mathfrak{A}_\ell) \leq 1$. Then we obtain

LEMMA 6.2 ([7, §6.2]). *Let $\mathfrak{A}_\ell$ be symmetric and positive definite. Then the Jacobi iteration (6.6) with*

$$\vartheta_\ell \leq 1/\rho(\mathfrak{D}_\ell^{-1} \mathfrak{A}_\ell) \quad \text{and} \quad \|\mathfrak{D}_\ell\|_2 \leq C_{\mathrm{sm}} \vartheta_\ell \|\mathfrak{A}_\ell\|_2 \tag{6.7}$$

satisfies the smoothing property with $\eta(\nu) = \eta_0(\nu)$ and $C_{\mathrm{sm}} \sigma_\ell = 1$. Moreover, this result holds for all symmetric and positive definite matrices $\mathfrak{D}_\ell$ satisfying the inequalities (6.7).

The Jacobi iteration (5.3) for the Sylvester equation with suitable θ satisfies the assumptions of the previous lemma (due to Lemma 5.2), therefore the smoothing property holds. In the case that the matrix $\mathfrak{D}_\ell$ is replaced by an approximation $\mathfrak{D}'_\ell$ due to the fact that we solve the diagonal Sylvester equation with a perturbed Cauchy matrix, we chose a symmetric $R(k)$-approximation of the Cauchy matrix. Since $\mathfrak{D}_\ell$ is well-conditioned, the approximation remains positive definite and satisfies the inequalities (6.7) with a possibly modified constant C_{sm}. Hence, also the Jacobi iteration with $R(k)$-Cauchy matrix approximation possesses the smoothing property. In combination with the approximation property from above, we obtain level-independent convergence of the W-cycle. Similarly, the V-cycle proof from [7] can be applied. The iterates X_i are all treated as full matrices and the multigrid method therefore has a complexity of $\mathcal{O}(n_\ell)$, where $X_i \in \mathbb{R}^{\sqrt{n_\ell} \times \sqrt{n_\ell}}$.

6.4. Truncation of the Iterates. The effect of the truncation in each step of the multigrid method (during the smoothing iteration and defect correction) can be regarded as an artificially limited machine precision. After one full multigrid cycle, the ith iterate $\mathfrak{x}_\ell^i$ on level ℓ is perturbed by $\mathfrak{s}_\ell^i$ so that the equation

$$\mathfrak{A}_\ell \mathfrak{x}_\ell^i = \mathfrak{b}_\ell - \mathfrak{A}_\ell \mathfrak{s}_\ell^i$$

holds. The vector $\mathfrak{s}_\ell^i$ accumulates all the perturbations of size ε during the ith cycle (due to the truncation to a fixed rank), $\|\mathfrak{s}_\ell^i\| \le C_{n_\ell}\varepsilon$. Since we expect a convergence rate of

$$\|\mathfrak{x}_\ell^i - \mathfrak{x}_\ell\| < \rho \|\mathfrak{x}_\ell^{i-1} - \mathfrak{x}_\ell\|, \qquad \rho < 1,$$

we immediately get

$$\|(\mathfrak{x}_\ell^i + \mathfrak{s}_\ell^i) - \mathfrak{x}_\ell\| \;<\; \rho\|\mathfrak{x}_\ell^{i-1} - \mathfrak{x}_\ell\| + C_{n_\ell}\varepsilon \;\le\; \underbrace{(\rho + \frac{C_{n_\ell}\varepsilon}{\|\mathfrak{x}_\ell^{i-1} - \mathfrak{x}_\ell\|})}_{\tilde\rho} \|\mathfrak{x}_\ell^{i-1} - \mathfrak{x}_\ell\|.$$

As long as we can represent $\mathfrak{x}_\ell^{i-1}$ sufficiently good, the term $C_{n_\ell}\varepsilon$ is small and the perturbed convergence rate $\tilde\rho \approx \rho$. As we come closer to the solution $\mathfrak{x}_\ell$ and fix the accuracy ε (the rank k, resp.), the convergence will break down (the iteration stagnates). This is also observed in the numerical tests.

REMARK 6.3 (non-symmetry). *Although the analysis given here requires the symmetry of the system matrix $\mathfrak{A}_\ell$ (which is induced by the symmetry of A and B), the multigrid method still works well for the non-symmetric case.*

In principle, the whole machinery of (algebraic) multigrid methods can be transferred to the Sylvester case by the tensor product relation. The crucial point is the connection between the hierarchies of the A matrices and the B matrices since these are generated independently. This "natural" anisotropy is considered in the next Section.

6.5. Anisotropic Case. As seen from (3.2) and the derived approximation, the stiffness matrix $\mathbb{A}$ is the sum $\mathbb{A}_x + \mathbb{A}_y$, where $\mathbb{A}_x, \mathbb{A}_y$ belong to the x- and y-variables. In the case of (3.2), the differential operator $\mathcal{A}[u](x,y)$ is $\mathcal{A} = -\Delta_x - \Delta_y$, i.e., the x- and y-parts are equal so that $\mathbb{A}_x$ and $\mathbb{A}_y$ have identical eigenvalues. A typical anisotropic differential operator is $\mathcal{A} = -\Delta_x - \varepsilon\Delta_y$ with small, positive ε. In this case the approximation property contains an additional factor $1/\varepsilon$, which may be formulated by the choice $\sigma_\ell = \varepsilon$. Therefore, we need a smoothing procedure such that the estimate (6.3) is improved by the same factor $\sigma_\ell = \varepsilon$. This can be obtained by the iteration

$$\mathcal{S}_\ell(\mathfrak{x}_\ell, \mathfrak{b}_\ell) = \mathfrak{S}_\ell \mathfrak{x}_\ell + \mathfrak{T}_\ell \mathfrak{b}_\ell \qquad \text{with } \mathfrak{S}_\ell = I - \mathfrak{A}_{x,\ell}^{-1}\mathfrak{A}_\ell, \quad \mathfrak{T}_\ell \mathfrak{b}_\ell = \mathfrak{A}_{x,\ell}^{-1}\mathfrak{b}_\ell, \tag{6.8}$$

where $\mathfrak{A}_\ell = \mathfrak{A}_{x,\ell} + \varepsilon\mathfrak{A}_{y,\ell}$. The terms $\mathfrak{A}_{x,\ell}$ and $\varepsilon\mathfrak{A}_{y,\ell}$ are the discretisations $\mathbb{A}_x, \mathbb{A}_y$ at level ℓ.

LEMMA 6.4 ([7, Lem 10.1.2]). *Let $\mathfrak{A}_{x,\ell}, \mathfrak{A}_{y,\ell}$ be symmetric and positive definite. In addition, we assume*

$$\mathfrak{A}_{x,\ell} \cdot \mathfrak{A}_{y,\ell} = \mathfrak{A}_{y,\ell} \cdot \mathfrak{A}_{x,\ell}, \qquad \|\mathfrak{A}_{y,\ell}\|_2 \le \mathit{const}\, \|\mathfrak{A}_\ell\|_2 .$$

Then the iteration (6.8) satisfies the smoothing property with $\eta(\nu) = \eta_0(\nu - 1)$ and $C_{\mathrm{sm}}\sigma_\ell = \varepsilon$.

Proof. For convenience, we repeat the proof, which is based on the identity

$$\begin{aligned}\mathfrak{A}_\ell \mathfrak{S}_\ell^\nu &= (\mathfrak{A}_{x,\ell} + \varepsilon \mathfrak{A}_{y,\ell})\, \mathfrak{S}_\ell^\nu = \mathfrak{A}_{x,\ell}\mathfrak{S}_\ell^\nu + \varepsilon \mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^\nu = \mathfrak{A}_{x,\ell}\left(I - \mathfrak{A}_{x,\ell}^{-1}\mathfrak{A}_\ell\right)\mathfrak{S}_\ell^{\nu-1} + \varepsilon \mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^\nu \\ &= (\mathfrak{A}_{x,\ell} - \mathfrak{A}_\ell)\, \mathfrak{S}_\ell^{\nu-1} + \varepsilon \mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^\nu = -\varepsilon \mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^{\nu-1} + \varepsilon \mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^\nu \\ &= \varepsilon \mathfrak{A}_{y,\ell}\,(I - \mathfrak{S}_\ell)\, \mathfrak{S}_\ell^{\nu-1}.\end{aligned}$$

The commutativity is used to show that $\mathfrak{S}_\ell$ is symmetric with eigenvalues in $[0,1]$. This implies $\|\,(I-\mathfrak{S}_\ell)\,\mathfrak{S}_\ell^{\nu-1}\|_2 = \rho((I-\mathfrak{S}_\ell)\,\mathfrak{S}_\ell^{\nu-1}) \le \eta_0(\nu-1)$ (cf. [8, Lemma 10.6.1]). The final result follows from $\|\mathfrak{A}_\ell\mathfrak{S}_\ell^\nu\|_2 = \varepsilon\|\mathfrak{A}_{y,\ell}\,(I-\mathfrak{S}_\ell)\,\mathfrak{S}_\ell^{\nu-1}\|_2 \le$ $\varepsilon\,\|\mathfrak{A}_{y,\ell}\|_2\,\|\,(I-\mathfrak{S}_\ell)\,\mathfrak{S}_\ell^{\nu-1}\|_2 \le \varepsilon\cdot const\cdot\|\mathfrak{A}_\ell\|_2\,\eta_0(\nu-1)$. ∎

Since the extra factor ε from the smoothing property cancels with the $1/\varepsilon$ factor in the approximation property, the estimate of the multigrid convergence rate is independent of ℓ and ε.

The commutativity $\mathfrak{A}_{x,\ell}\cdot\mathfrak{A}_{y,\ell} = \mathfrak{A}_{y,\ell}\cdot\mathfrak{A}_{x,\ell}$ holds in the case of the Sylvester equation because of the tensor structure.

The solution of the system $\mathfrak{A}_{x,\ell}$ in each step of the iteration (6.8) requires the solution of a system

$$A_\ell X_\ell = C_\ell, \qquad X_\ell \in \mathbb{R}^{\sqrt{n_\ell}\times\sqrt{n_\ell}},$$

which can be done columnwise for full matrices or just for the k column vectors of U in the $R(k)$-matrix representation of Definition 1.2.

6.6. Weaker Regularity. In the case of a re-entrant corner of Ω, the full regularity is not satisfied, but $\mathcal{A} : H^{-s}(\Omega\times\Omega) \to H^{2-s}(\Omega\times\Omega)$ is an isomorphism for some $s\in[0,1)$ (cf. [9]). In this case, one has to modify the matrix norms in (6.3) and (6.5):

$$\left\|\mathfrak{A}_\ell^{1-s/2}\mathfrak{S}_\ell^\nu\,\mathfrak{A}_\ell^{-s/2}\right\|_2 \le C_{\rm sm}\,\|\mathfrak{A}_\ell\|_2^{1-s}\,\sigma_\ell\eta(\nu), \tag{6.9a}$$

$$\|\mathfrak{A}_\ell^{-s/2}\left(\mathfrak{A}_\ell^{-1} - p\mathfrak{A}_{\ell-1}^{-1}r\right)\mathfrak{A}_\ell^{s/2}\|_2 \le C_{\rm app}/\left(\|\mathfrak{A}_\ell\|_2^{1-s}\,\sigma_\ell\right), \tag{6.9b}$$

involving fractional powers of $\mathfrak{A}_\ell$. In the case of $\mathfrak{A}_{x,\ell}\cdot\mathfrak{A}_{y,\ell} = \mathfrak{A}_{y,\ell}\cdot\mathfrak{A}_{x,\ell}$, it follows that $\mathfrak{A}_\ell^\alpha$ $(0\le\alpha\le 1)$ with $\mathfrak{A}_\ell = \mathfrak{A}_{x,\ell} + \varepsilon\mathfrak{A}_{y,\ell}$ is spectrally equivalent with $\mathfrak{A}_{x,\ell}^\alpha + \varepsilon^\alpha\mathfrak{A}_{y,\ell}^\alpha$. Using these matrices in (6.9a,b) instead of $\mathfrak{A}_\ell^\alpha$, one can repeat the proof of Lemma 6.4 with $\sigma_\ell = \varepsilon^{1-s}$ and $\eta(\nu) = (\eta_0(\nu))^{1-s}$.

7. Numerical Results. In this section we apply the $R(k)$-multigrid algorithm developed in the previous sections to the model problem of Section 1.1 with $\omega := 1$ in (1.1), namely the first step of Newtons method where we have to solve (1.6) with a rank one right-hand side C and the two-dimensional discrete Laplacian A. The simple geometry allows us to use two-dimensional bilinear (tensor) basis functions ϕ_i and a nested hierarchy of grids with a coarse grid that contains $n_0 = 9$ degrees of freedom, see Figure 7.1.

The computations are performed on a SUN ULTRASPARC III with 900 MHz CPU clock rate and 150 MHz memory clock rate. We make use of the LAPACK and BLAS libraries (`http://www.netlib.org`) for the truncation procedure and use the standard C programming language otherwise.

The initial approximation on level ℓ is obtained by prolongation of a solution from level $\ell-1$, i.e., we use a nested iteration so that only $\mathcal{O}(1)$ steps on the fine grid are necessary in order to reduce the error down to the size of the discretisation error. The

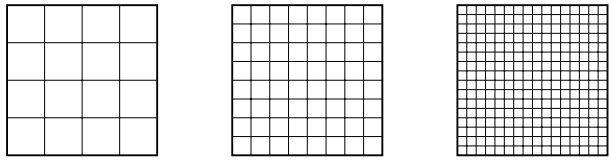

FIG. 7.1. *The three coarsest grids with $n_0 = 9, n_1 = 49$ and $n_2 = 225$ interior nodes.*

rank on level $\ell - 1, \dots, 0$ is chosen as twice the rank k on the fine grid ℓ on which we want to compute the solution. In the V-cycle multigrid we use $\nu = 2$ pre- and postsmoothing steps.

7.1. Discretisation Error. First, we perform the multigrid method with rank $k = 20$ in order to produce a reference solution on each level $\ell = 0, \dots, 9$. The solution on level $\ell = 9$ is used to estimate the discretisation error on level $\ell = 0, \dots, 8$, see Table 7.1. The rank k is large enough so that the truncation has no influence.

$\ell = 3$	$\ell = 4$	$\ell = 5$	$\ell = 6$	$\ell = 7$	$\ell = 8$
$n = 31^2$	$n = 63^2$	$n = 127^2$	$n = 255^2$	$n = 511^2$	$n = 1023^2$
5.1×10^{-2}	2.0×10^{-2}	7.1×10^{-3}	2.5×10^{-3}	8.9×10^{-4}	1.9×10^{-4}

TABLE 7.1
The relative discretisation error $\|\mathfrak{x}_\ell - \mathfrak{x}_9\|/\|\mathfrak{x}_9\|$ in the L^2-norm.

Alternatively, one could use the multigrid method without truncation working with full matrices instead of the $R(k)$-matrix format. For level $\ell = 3$ this takes only 2.6 seconds to compute an accurate solution, but the complexity is quadratic in n so that on level $\ell = 5$ we need more that 600 seconds for the solution and on level $\ell = 8$ we would (theoretically) need approximately 775 hours.

7.2. Truncation Error. Next, we perform the multigrid cycle on level $\ell = 6$ with fixed rank $k = 2$, so that the truncation error ε due to the small rank k becomes dominant, see Table 7.2. As was expected from the theory, the convergence breaks down after we get close to the solution. Since we are already at the size of the discretisation error, we can stop the iteration after three steps which takes 9.7 seconds.

$i = 0$	$i = 1$	$i = 2$	$i = 3$	$i = 4$
6.4×10^{-3}	4.2×10^{-3}	2.8×10^{-3}	2.0×10^{-3}	2.0×10^{-3}

TABLE 7.2
The relative error $\|\mathfrak{x}_\ell^i - \mathfrak{x}_\ell\|/\|\mathfrak{x}_\ell\|$ on level $\ell = 6$ in the L^2-norm for the iterates $i = 0, \dots, 4$.

7.3. Large Scale Problems. The last three levels $\ell = 7, 8, 9$ in our numerical test would require to store (and compute) solution matrices X_ℓ of size up to 4190209×4190209. Without the low rank format the storage in double precision of these requires 128 Terabyte. In the $R(k)$-matrix format we need only 256 MB. In the following numerical example we use the $R(k)$-multigrid algorithm based on the $R(k)$-Richardson iteration. The time in seconds for the computation of a solution that is accurate up to the discretisation error is given in Table 7.3.

The nested iteration combined with the multigrid method has a complexity of $\mathcal{O}(k^2 n_\ell)$ to solve the discrete system $\mathfrak{A}_\ell \mathfrak{x}_\ell = \mathfrak{b}_\ell$ on level ℓ. Although the dependency

ℓ	k	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$	time
7	3	2.3×10^{-3}	1.5×10^{-3}	9.1×10^{-4}	5.4×10^{-4}		84.2
8	4	9.6×10^{-4}	6.5×10^{-4}	4.4×10^{-4}	3.0×10^{-4}	2.1×10^{-4}	1064.1
9	4	3.5×10^{-4}	2.4×10^{-4}	1.6×10^{-4}	1.1×10^{-4}	7.0×10^{-5}	6964.1

TABLE 7.3

The relative error $\|\mathfrak{x}_\ell^i - \mathfrak{x}_\ell\|/\|\mathfrak{x}_\ell\|$ on level $\ell = 7, 8, 9$ in the L^2-norm for the iterates $i = 0, \ldots, 4$.

is linear in n_ℓ, the rank k for the representation of the solution depends on n_ℓ, typically $k = \log(n_\ell)$. Therefore, the overall complexity of our algorithm is $\mathcal{O}(n_\ell \log^2(n_\ell))$. A goal for future research is to reduce this complexity down to $\mathcal{O}(n_\ell)$.

7.4. Second Model Problem. For the second model problem from Section 1.2 we consider the dependency on the parameter β that governs the non-symmetry of the system. For $\beta = 0$ the model problem is identical to the one considered in the previous section.

For larger β we face two distinct problems: first, the required rank for the accurate representation of the solution will increase, because the spectrum of the system matrix A will become complex and approach the imaginary axis. Second, the convergence rate of the multigrid method will not be bounded away from 1 as $\beta \to \infty$, because the smoother (in this case $R(k)$-Richardson) is not suitable for convection-dominated problems.

k	$\beta = 1$	$\beta = 10$	$\beta = 100$
1	2.6×10^{-2}	6.3×10^{-2}	1.0×10^{-1}
2	2.3×10^{-3}	1.2×10^{-2}	3.1×10^{-2}
3	4.2×10^{-4}	2.4×10^{-3}	1.3×10^{-2}
4	1.2×10^{-4}	6.8×10^{-4}	6.1×10^{-3}
5	3.2×10^{-5}	1.8×10^{-4}	3.0×10^{-3}
6	7.5×10^{-6}	5.2×10^{-5}	1.6×10^{-3}
7	1.9×10^{-6}	1.6×10^{-5}	8.1×10^{-4}
8	4.9×10^{-7}	5.0×10^{-6}	4.3×10^{-4}
9	1.5×10^{-7}	1.6×10^{-6}	2.2×10^{-4}

TABLE 7.4

The first $k = 1, \ldots, 9$ singular values of the solution X_ℓ on level $\ell = 5$ for the parameter $\beta \in \{1, 10, 100\}$.

In Table 7.4 we can clearly see that for $\beta = 100$ the decay of the singular values of the solution is less steep than for $\beta = 1$. In the multigrid method we use the damping parameter $\theta := 1/\|\mathfrak{A}_\ell\|_2$, but the coarsest grid level will now depend on the parameter β: for $\beta = 1$ we choose the coarsest grid level $\ell = 0$, and for $\beta = 20, 40, 80$ we take $\ell = 1, 2, 3$, so that the ratio $\beta \cdot h_\ell$ is constant on the coarsest grid. Of course, for larger values of β the coarsest grid on which we have to solve the Sylvester equation by some other means will grow. The convergence rates of the multigrid iteration are given in Table 7.5. If either the damping parameter θ is chosen too large or the coarsest grid too coarse, then the multigrid iteration diverges.

7.5. First Model Problem. At last we consider the first model problem from Section 1.1 (parameter $\kappa(\xi) = 10000$ for $\xi \in (\frac{1}{2}, 1) \times (0, 1)$), where a Riccati equation has to be solved. In each step of Newton's method (initial guess $X_{\ell-1}$ from

i	$\beta = 1$		$\beta = 20$		$\beta = 40$		$\beta = 80$	
1	1.3×10^{-2}	.55	1.3×10^{-2}	.44	1.9×10^{-2}	.43	2.4×10^{-2}	.41
2	7.1×10^{-3}	.55	7.5×10^{-3}	.57	1.0×10^{-2}	.54	1.4×10^{-2}	.58
3	4.1×10^{-3}	.57	2.2×10^{-3}	.29	3.6×10^{-3}	.35	5.6×10^{-3}	.40
4	2.4×10^{-3}	.58	1.2×10^{-3}	.55	3.0×10^{-3}	.83	3.4×10^{-3}	.60
5	1.4×10^{-3}	.59	5.6×10^{-4}	.46	8.5×10^{-4}	.29	2.0×10^{-3}	.58
6	8.1×10^{-4}	.59	2.5×10^{-4}	.45	4.5×10^{-4}	.53	1.8×10^{-3}	.91
7	4.8×10^{-4}	.59	8.8×10^{-5}	.35	2.4×10^{-4}	.52	1.3×10^{-3}	.74

TABLE 7.5

The relative error $\|\mathfrak{x}_\ell^i - \mathfrak{x}_\ell\|/\|\mathfrak{x}_\ell\|$ on level $\ell = 5$ in the L^2-norm and parameter $\beta \in \{1, 20, 40, 80\}$ (left: relative error; right: convergence rate).

the coarse grid) we have to solve a Lyapunov equation which is done by using the multigrid method. Here, we employ the Jacobi smoother, where the damping factor is computed as in Section 5 for the coarsest grid solver on level $\ell = 1$, i.e., $\theta := 2/\|\mathrm{diag}(\mathfrak{A}_\ell)^{-1/2}\ \mathfrak{A}_\ell\ \mathrm{diag}(\mathfrak{A}_\ell)^{-1/2}\|_2$. In the first three steps of the multigrid method we use the same choice of the damping parameter. From step four on we use $\theta := 1/2$. The Cauchy matrix approximation $\tilde{\mathcal{C}}$ uses a rank of 1. We fix the discretisation level $\ell = 5$ with $n_\ell = 16129$ degrees of freedom and a solution matrix $X \in \mathbb{R}^{n_\ell \times n_\ell}$. The rank for the $R(k)$-multigrid algorithm is fixed to $k = 20$. We perform three Newton steps and apply $i = 10$ multigrid steps each.

By X^j we denote the (almost) exact solution of the Lyapunov equation in the j-th Newton step on level ℓ. By X_i^j we denote the i-th iterate of the multigrid iteration in the j-th Newton step. In Table 7.6 the relative error $\|X^j - X_i^j\|_2/\|X^j\|_2$ is reported for the three Newton steps $j = 1, 2, 3$.

i	NS $j = 1$		NS $j = 2$		NS $j = 3$	
1	6.1×10^{-3}	.13	7.5×10^{-6}	.08	3.3×10^{-7}	.74
2	7.8×10^{-3}	.13	5.1×10^{-6}	.67	2.4×10^{-7}	.74
3	1.3×10^{-4}	.16	3.8×10^{-6}	.75	1.8×10^{-7}	.74
4	7.4×10^{-5}	.58	2.8×10^{-6}	.74	1.3×10^{-7}	.74
5	5.1×10^{-5}	.69	2.1×10^{-6}	.74	9.8×10^{-8}	.74
6	3.6×10^{-5}	.71	1.5×10^{-6}	.74	7.3×10^{-8}	.74
7	2.6×10^{-5}	.73	1.1×10^{-6}	.74	5.4×10^{-8}	.74
8	1.9×10^{-5}	.73	8.3×10^{-7}	.74	4.0×10^{-8}	.74
9	1.4×10^{-5}	.73	6.1×10^{-7}	.74	2.9×10^{-8}	.74
10	1.0×10^{-5}	.73	4.5×10^{-7}	.74	2.1×10^{-8}	.74

TABLE 7.6

Convergence rates for the first $i = 1, \ldots, 10$ steps of the multigrid iteration in the Newton step $j = 1, 2, 3$, based on the Jacobi smoother (left: relative error; right: convergence rate).

We conclude that the $R(k)$-multigrid method is well-suited for the solution of the linear matrix equations in each step of Newton's method to solve the algebraic matrix Riccati equation. The Jacobi smoother yields uniformly bounded convergence rates $\rho \approx 0.74$. If the parameter κ is much smaller, i.e. $\kappa(\xi) = \mathcal{O}(1)$, then the convergence rate is $\rho \approx 0.52$.

REFERENCES

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

[2] A. Antoulas, D. Sorensen: *The Sylvester equation and approximate balanced reduction.* Lin. Alg. Appl. 351–352, 671–700, 2002.

[3] R. H. Bartels, G. W. Stewart: *Solution of the matrix equation* $AX + XB = C$. Comm. ACM 15, 820–826 (1972).

[4] H. Banks, K. Kunisch: *The linear regulator problem for parabolic systems.* SIAM J. Contr. Opt. 22, 684–696, 1984.

[5] G. H. Golub, C. F. Van Loan: *Matrix computations.* Johns Hopkins University Press, London, 1996.

[6] L. Grasedyck: *Existence of a low rank or* $\mathcal{H}$*-matrix approximant to the solution of a Sylvester equation.* Num. Lin. Alg. Appl. 11, 371–389 (2004).

[7] W. Hackbusch. *Multi-grid methods and applications*. Springer-Verlag, Berlin, 2nd edition, 2003.

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

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

[10] W. Hackbusch, S. Sauter: *Composite finite elements for the approximation of PDEs on domains with complicated micro-structures.* Numer. Math. 75, 447–472 (1997).

[11] D. Kleinman: *On an iterative technique for Riccati equations computation.* IEEE Trans. Aut. Contr.13, 114–115, 1968.

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

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

[14] R. Lezius R, F. Tröltzsch: *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.

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

[16] T. Penzl: *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case.* Sys. Contr. Lett. 40, 139–144, 2000.

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

[18] T. Penzl: *A cyclic low rank Smith method for large sparse Lyapunov equations.* SIAM J. Sci. Comput. 21, 1401–1418 (2000).

[19] J. Ruge, K. Stüben: *Efficient solution of finite difference and finite element equations by algebraic multigrid.* in Paddon, D.J., and Holstein, H., (eds.): Multigrid methods for integral and differential equations, Oxford Univ. Press., N.Y. (1985), pp. 169–212.

[20] D. Hu, L. Reichel: *Krylov-subspace methods for the Sylvester equation.* Lin. Alg. Appl. 172, 283–313 (1992).

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