ELSEVIER
Theoretical Computer Science 259 (2001) 623–638
Theoretical Computer Science
www.elsevier.com/locate/tcs

Fast algorithms for the Sylvester equation $AX - XB^{\mathrm{T}} = C$

Peter Kirrinnis

Universität Bonn, Institut für Informatik II, Römerstr. 164, D-53117 Bonn, Germany

Received 23 August 1999; revised 26 May 2000; accepted 18 July 2000
Communicated by V. Pan

Abstract

For given matrices $A \in F^{m\times m}$, $B \in F^{n\times n}$, and $C \in F^{m\times n}$ over an arbitrary field F, the matrix equation $AX - XB^{\mathrm{T}} = C$ has a unique solution $X \in F^{m\times n}$ if and only if A and B have disjoint spectra. We describe an algorithm that computes the solution X for $m, n \leqslant N$ with $\mathrm{O}(N^{\beta} \cdot \log N)$ arithmetic operations in F, where $\beta > 2$ is such that $M \times M$ matrices can be multiplied with $\mathrm{O}(M^{\beta})$ arithmetic operations, e.g., $\beta = 2.376$. It seems that before no better bound than $\mathrm{O}(m^3 \cdot n^3)$ arithmetic operations was known. The state of the art in numerical analysis is $\mathrm{O}(n^3 + m^3)$ flops, but these algorithms (due to Bartels/Stewart and Golub/Nash/van Loan) involve Schur decompositions, i.e., they compute the eigenvalues of at least one of A and B, and can hence not be transferred for general F. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Matrix equations; Sylvester equation; Fast algorithms; Algebraic complexity

1. Introduction

The inhomogeneous linear matrix equation

$$AX - XB^{\mathrm{T}} = C \quad (A \in F^{m\times m}, B \in F^{n\times n}, C, X \in F^{m\times n}) \tag{1.1}$$

is called the Sylvester equation (over the field F). It has a unique solution X if and only if the coefficient matrices A and B have no eigenvalues in common [18, Theorem 4.4.6, 16, Section 15.1].

The Sylvester equation appears in various branches of mathematics, in most cases for $F = \mathbb{R}$ or $F = \mathbb{C}$. We recall some special cases — not all with unique solutions — from [16, Chapter 15]: matrix inversion ($AX = I$), linear systems ($AX = C$), computation of eigenvectors ($AX - X\Lambda = 0$ with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$), and commuting matrices ($AX - XA = 0$).

E-mail address: kirr@cs.uni-bonn.de (P. Kirrinnis).

PII: S0304-3975(00)00322-4

Another example is block diagonalization of block triangular matrices, also mentioned in [16, Chapter 15]: the matrix

$$\begin{pmatrix} I & -X \\ 0 & I \end{pmatrix} \begin{pmatrix} A & -C \\ 0 & B^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} I & X \\ 0 & I \end{pmatrix}^{-1} = \begin{pmatrix} A & AX - XB^{\mathrm{T}} - X \\ 0 & B^{\mathrm{T}} \end{pmatrix}$$

is block diagonal iff X solves the Sylvester equation $AX - XB^{\mathrm{T}} = C$. The application of the Sylvester equation for block diagonalization of real Schur canonical forms is described in [13, Section 7.6.3].

The Sylvester equation plays an important role in control theory. An important special case is the Lyapunov equation $AX + XA^{\mathrm{T}} = C$. Another application is solving the nonlinear matrix equation

$$AX - XB^{\mathrm{T}} = C + XDX$$

called the *algebraic Riccati equation*, which arises naturally in systems and control theory. Some references to the corresponding literature are given in the introduction of [3]. In the control theory literature, Riccati equations are solved by reduction to eigenvector problems [3, 22]. The other way round, several methods for refining approximations of invariant subspaces can be reduced to the Riccati equation [9]. A step of the direct iteration $AX_{k+1} - X_{k+1}B^{\mathrm{T}} = C + X_kDX_k$ for solving the Riccati equation is obviously a Sylvester equation. Newton iteration for the Riccati equation also leads to a Sylvester equation. Both algorithms are analyzed in [9, Section 4].

As a last application of the Sylvester equation we mention the solution of discretized elliptic boundary problems on rectangular domains [24]. In [24], an iterative algorithm for solving the Sylvester equation is proposed.

The standard direct algorithms for the Sylvester equation with matrices over the reals are the Bartels–Stewart algorithm [5] (see [13, Section 7.6.2] for a simplified version) and a Hessenberg–Schur method proposed by Golub et al. [14]. In the words of the latter authors, “the crux of the Bartels–Stewart algorithm” is that the first step is to compute real Schur decompositions of A and B, i.e., orthogonal matrices U and V such that $U^{\mathrm{T}}AU$ and $V^{\mathrm{T}}BV$ are quasi-triangular (i.e., triangular up to some 2×2 blocks on the diagonal, which originate from pairs of complex conjugate eigenvalues). For complex matrices, U and V are unitary and the transformations yield triangular matrices. The LAPACK [2] algorithms for the Sylvester equation also require both the matrices A and B to be reduced to Schur form.

The Hessenberg–Schur method [14] requires only one of the matrices A or B to be reduced to Schur form, while for the other one reduction to Hessenberg form is sufficient. This reduces the operation count (measured in flops) by a constant factor. Both the Bartels–Stewart and the Golub–Nash–Van Loan algorithm use $\mathrm{O}(m^3 + n^3)$ floating point operations, if one assumes that an $M \times M$ matrix can be reduced to Schur form with $\mathrm{O}(M^3)$ operations. More precise bounds are given in [5, 14].

As the Sylvester equation is a linear equation, the entries of the solution X are rational expressions in the entries of A, B, and C. Opposed to that, the entries of a

Schur form of the matrix A (i.e., the matrix $U^{\mathrm{T}}AU$) need not be rational expressions in the entries of A, B, and C. If for instance all entries of A, B, and C are rational numbers, then so are the entries of X, while the diagonal elements of a Schur form of A are the eigenvalues of A, which need not be rational.

In numerical analysis, Schur decompositions are usually computed in two steps: first, A is transformed into Hessenberg form with Householder transforms, and then QR iteration is used to get rid of the subdiagonal. Hence, there are intermediate results that are not computed exactly, but approximated by an iterative numerical process. This is appropriate for numerical analysis, but it raises the question whether there is a competitive (say, $\mathrm{O}(m^3+n^3)$) algorithm that uses only rational operations and can hence be used for arbitrary fields.

A brute force attack in this direction is to rewrite the Sylvester equation in matrix vector form, using the Kronecker Product

$$U \otimes V = (u_{ij}V)$$

and the vec operator, which maps an $m \times n$ matrix to the $m \cdot n$ vector of its columns (see [18, Chapter 4]). With the relation $\mathrm{vec}(AXB) = (B^{\mathrm{T}} \otimes A)\mathrm{vec}(X)$, the Sylvester equation $AX - XB^{\mathrm{T}} = C$ can be written as

$$(I_n \otimes A - B \otimes I_m)\mathrm{vec}(X) = \mathrm{vec}(C). \tag{1.2}$$

This $mn \times mn$ system can be solved by Gaussian elimination with $\mathrm{O}(m^3n^3)$ flops, but this is unacceptable. In [24], it is proposed to solve (1.2) numerically by an iterative algorithm. So the second question is whether the special structure of the large linear system (1.2) can be exploited using only rational operations.

A third question is whether asymptotically fast matrix multiplication algorithms can be used to solve the Sylvester equation. It is now accepted that Strassen's $\mathrm{O}(M^{2.81})$ algorithm [26] and Winograd's variant [27] for multiplying and inverting $M \times M$ matrices are of practical relevance, see, e.g. [16, Chapter 22], [15], [4]. Although the present paper does not discuss practical issues like numerical stability, there is hope that exploiting fast matrix multiplication for the Sylvester equation will eventually also be useful for numerical analysis.

Throughout this paper, F denotes a field, and $\beta > 2$ is such that $M \times M$ matrices over F can be multiplied with $\mathrm{O}(M^{\beta})$ arithmetic operations in F. According to the best complexity bound currently known — due to Coppersmith and Winograd [8] — we can choose $\beta < 2.376$ for any field F. The assumption $\beta > 2$ has technical reasons. The case $\beta = 2$ brings in additional logarithmic factors in the time bounds and must be dealt with separately in the proofs, see the remark at the bottom of p. 310 in [20].

The complexity bounds in this paper refer to the total complexity of algebraic computation trees as defined in [6, Section 4.4]. We refer to the complexity as to "time" or the "number of arithmetic operations" (i.e., $+$, $-$, $*$, $/$), although tests for equality are counted, too. Our main result is the following complexity bound:

Theorem 1.1. *Let $A \in F^{m\times m}$ and $B \in F^{n\times n}$ with disjoint spectra, let $C \in F^{m\times n}$, and let $m, n \leqslant N$. Then the (unique) solution X of the Sylvester equation $AX - XB^{\mathrm{T}} = C$ can be computed with $\mathrm{O}(N^{\beta} \cdot \log N)$ arithmetic operations.*

If $M \times M$ matrices can be inverted with $\mathrm{O}(M^{\beta})$ operations, then $\mathrm{O}(M^{\beta})$ is also an upper bound for the complexity of $M \times M$ matrix multiplication. As matrix inversion is a special case of the Sylvester equation, the time bound in Theorem 1.1 is sharp up to the logarithmic factor.

Note that Theorem 1.1 does not apply to the particularly interesting and important case of eigenvector computation, because we have assumed that A and B have disjoint spectra.

2. Outline of the algorithm

The idea of the algorithm is to transform the coefficient matrices A and B to a special form for which the Sylvester equation can be solved easily. The Sylvester equation is invariant w.r.t. similarity transforms in the following sense:

Lemma 2.1. *Let $A \in F^{m\times m}$, $B \in F^{n\times n}$, and $C \in F^{m\times n}$. Let $U \in \mathrm{GL}(m)$ and $V \in \mathrm{GL}(n)$ and define $A' = U^{-1}AU$ and $B' = V^{-1}BV$. Then $Y \in F^{m\times n}$ solves the equation $A'Y - Y(B')^{\mathrm{T}} = U^{-1}C(V^{-1})^{\mathrm{T}}$ if and only if $X = UYV^{\mathrm{T}}$ solves $AX - XB^{\mathrm{T}} = C$.*

The Bartels–Stewart and Golub–Nash–Van Loan algorithms are both based on this lemma, and so is the one presented here.

The characteristic polynomial of a matrix $U \in F^{k\times k}$ is denoted by $\chi_U(z) = \det(z \cdot I_k - U)$. First, we show that if $B = (b_{i,j})$ is an upper Hessenberg matrix (i.e., $b_{i,j} = 0$ for all $i > j+1$) with all subdiagonal entries $b_{i+1,i}$ equal to 1, then the last column x_n of the solution X fulfills the linear equation $\chi_B(A)x_n = d$, where $d \in F^m$ can be computed from A, B, and C. Once x_n is known, the other columns of X can be computed from (1.2) by backward substitution. The Sylvester equation for Hessenberg matrices is discussed in detail in Section 3.

If B is a companion or Frobenius matrix, i.e.,

$$B = \begin{pmatrix} 0 & 0 \cdots 0 & -b_0 \\ 1 & 0 \cdots 0 & -b_1 \\ 0 & 1 \cdots 0 & -b_2 \\ \vdots & \ddots & \vdots \\ 0 & 0 \cdots 1 & -b_{n-1} \end{pmatrix},$$

then the characteristic polynomial of B is $\chi_B(z) = z^n + b_{n-1}z^{n-1} + \cdots + b_1 z + b_0$. If A is also a Frobenius matrix, then the equation $\chi_B(A)x_n = d$ can be translated into a polynomial equation that can be solved by polynomial GCD computation. Moreover, the r.h.s. d can be computed efficiently and the backward substitution for the other columns of X can be performed fast due to the sparsity of A and B. In Section 4, we show that for Frobenius matrices $A \in F^{m\times m}$ and $B \in F^{n\times n}$ with disjoint spectra, the Sylvester equation $AX - XB^{\mathrm{T}} = C$ can be solved with $\mathrm{O}(m \cdot n)$ operations.

The general case is reduced to the Frobenius matrix case by Keller-Gehrig's [20] asymptotically fast version of the Krylov method for computing the characteristic polynomial [11, Section 7.8]. In the generic case, i.e., if the entries of A and B are indeterminates, this algorithm produces nonsingular matrices U and V such that $A' = U^{-1}AU$ and $B' = V^{-1}BV$ are Frobenius matrices. This reduction is dealt with in Section 5.

In general Keller-Gehrig's algorithm produces matrices A' and B' that are upper block triangular and have Frobenius matrices on the block diagonal. For such matrices, the Sylvester equation is solved by reduction to the Frobenius matrix case with a divide and conquer algorithm. This algorithm is described in Section 6. Its analysis yields a proof for Theorem 1.1.

3. The Sylvester equation for Hessenberg matrices

In this section, we show that if in the Sylvester equation $AX - XB^{\mathrm{T}} = C$ the matrix B is an upper Hessenberg matrix with all subdiagonal entries equal to 1, then the last column x_n of the solution X fulfills the linear equation $p(A)x_n = d$ for some $d \in F^m$, where p is the characteristic polynomial of B. Once x_n is known, the other columns of X can be computed by backward substitution from (1.2). The relations derived here give rise to an efficient algorithm for the case where both A and B are Frobenius matrices, see Section 4. The results hold for matrices over an arbitrary field F.

The equation for x_n is derived by "inserting" the matrix A into a polynomial equation. "Inserting" a matrix $A \in F^{m\times m}$ into a matrix polynomial $P(z) \in F[z]^{n\times n}$ is meant in the sense of generalizing the concept of Kronecker product:

Lemma 3.1. *Let $A \in F^{m\times m}$. Then*

$$\begin{aligned} \eta_A : F[z]^{n\times n} &\to (F^{m\times m})^{n\times n} \\ (p_{i,j}(z))_{1\leqslant i,j\leqslant n} &\mapsto (p_{i,j}(A))_{1\leqslant i,j\leqslant n} \end{aligned}$$

defines a homomorphism of F-algebras. In particular, $\eta_A(B) = B \otimes I_m$ for $B \in F^{n\times n}$ and $\eta_A(z^k \cdot I_n) = I_n \otimes A^k$ for $k \in \mathbb{N}$.

This concept should not be mixed up with another standard concept of inserting a matrix into a matrix polynomial for the special case $m = n$, namely $\sum_{k=0}^{d} P_k z^k \mapsto \sum_{k=0}^{d} P_k A^k$ for coefficient matrices $P_0, \ldots, P_d \in F^{n\times n}$.

In the following, R denotes a commutative ring with unity, e.g., the ring $F[z]$ of univariate polynomials over F.

Definition 3.2. Let $B=(b_{i,j})\in R^{n\times n}$. For $1\leqslant i,\ j\leqslant n$, let $B\langle i,j\rangle \in R^{(n-1)\times(n-1)}$ denote the matrix defined by deleting the ith row and the jth column of B. The cofactor of $b_{i,j}$ is $y_{j,i}=(-1)^{i+j}\cdot \det B\langle i,j\rangle$. The matrix $Y=(y_{i,j})_{1\leqslant i,j\leqslant n}$ is called the adjoint of B.

The following fact can be found in almost all textbooks on linear algebra, e.g., [17, 0.8.2]:

Lemma 3.3. *Let $B\in R^{n\times n}$, and let Y be its adjoint. Then $Y\cdot B=B\cdot Y=\det B\cdot I_n$.*

Lemma 3.4. *Let $B\in R^{n\times n}$ be upper Hessenberg with all subdiagonal elements equal to -1. Let $B_k=(b_{i,j})_{1\leqslant i,j\leqslant k}$. Then (with the notation of Definition 3.2) the entries of the last row of Y are $y_{n,1}=1$ and $y_{n,k}=\det B_{k-1}$ for $k>1$.*

Proof. The matrix $B\langle k,n\rangle$ has the form

$$\begin{pmatrix} B_{k-1} & C \\ 0 & D \end{pmatrix},$$

where D is an upper triangular $(n-k)\times(n-k)$ matrix with all diagonal elements equal to -1. Hence

$$\begin{aligned} y_{n,k} &= (-1)^{k+n}\cdot\det B\langle k,n\rangle = (-1)^{k+n}\cdot\det B_{k-1}\cdot\det D \\ &= (-1)^{k+n}\cdot\det B_{k-1}\cdot(-1)^{n-k} = \det B_{k-1}. \qquad \square \end{aligned}$$

Corollary 3.5. *Let $B\in F^{n\times n}$ be upper Hessenberg with all subdiagonal elements equal to 1. Then the last row of the adjoint $Y(z)$ of zI_n-B is*

$$(1,\chi_{B_1}(z),\chi_{B_2}(z),\ldots,\chi_{B_{n-1}}(z)). \tag{3.1}$$

Proof. Apply Lemma 3.4 to $R=F[z]$ and zI_n-B in place of B. $\square$

Lemma 3.6. *Let B be as in Corollary 3.5. Let $A\in F^{m\times m}$, $C\in F^{m\times n}$ with columns $c_1,c_2,\ldots,c_n$, and let $X\in F^{m\times n}$ be a solution to the Sylvester equation $AX-XB^{\mathrm{T}}=C$. Then the last column x_n of X fulfils*

$$\chi_B(A)\cdot x_n=d,$$

where (with $\chi_{B_0}(z)=1$)

$$d=\sum_{k=0}^{n-1}\chi_{B_k}(A)\cdot c_{k+1}.$$

Proof. Let $Y(z)\in F[z]^{n\times n}$ be the adjoint of zI_n-B. Then

$$Y(z)\cdot(zI_n-B)=\chi_B(z)\cdot I_n \tag{3.2}$$

in $F[z]^{n\times n}$ because of Lemma 3.3. The last row of $Y(z)$ is given by (3.1). Applying the F-algebra-homomorphism η_A (Lemma 3.1) to (3.2) yields

$$\eta_A(Y)\cdot(I_n\otimes A - B\otimes I_m) = I_n\otimes\chi_B(A).$$

With the vectorized form (1.2) of the Sylvester equation, this implies

$$\eta_A(T)\cdot\operatorname{vec}(C) = (I_n\otimes\chi_B(A))\cdot\operatorname{vec}(X).$$

This is an equation in $(F^m)^n$. On the r.h.s., the vector of the last m entries is d because of (3.1), and on the l.h.s. it is $\chi_B(A)\cdot x_n$. □

Note that $\chi_B(A)$ is nonsingular (and hence x_n is uniquely determined) if and only if A and B have no eigenvalues in common. In any case, the other columns $x_1,\dots,x_{n-1}$ of X are determined by x_n and can be computed by backward substitution in (1.2):

Lemma 3.7. *Let A, B, C, X be as in Lemma* 3.6. *Then the first $n-1$ columns $x_1,\dots,x_{n-1}$ of X are given recursively by*

$$x_{k-1} = Ax_k - b_{k,k}x_k - b_{k,k+1}x_{k+1} - \cdots - b_{k,n}x_n - c_k \quad \textit{for } n\geqslant k\geqslant 2.$$

Proof. This follows from a closer look at (1.2) for this special case ($I=I_m$):

$$\begin{pmatrix} A-b_{1,1}I & -b_{1,2}I & -b_{1,3}I & \cdots & -b_{1,n}I \\ -I & A-b_{2,2}I & -b_{2,3}I & \cdots & -b_{2,n}I \\ & -I & A-b_{3,3}I & \cdots & -b_{3,n}I \\ & & \ddots & \ddots & \dots \\ & & & -I & A-b_{n,n}I \end{pmatrix}\begin{pmatrix} x_1\\ x_2\\ x_3\\ \vdots\\ x_n\end{pmatrix} = \begin{pmatrix} c_1\\ c_2\\ c_3\\ \vdots\\ c_n\end{pmatrix}. \qquad \square$$

4. Fast algorithms for Frobenius matrices

If A is a Frobenius matrix, then computing $q(A)$ for a polynomial q can be reduced to polynomial multiplication and division, and linear equations of the form $q(A)x=d$ (like in Lemma 3.6) can be solved by polynomial GCD algorithms. This section shows how to exploit relations of this type to solve the Sylvester equation for Frobenius matrices A and B efficiently.

Let us first recall some results about polynomial arithmetic. Let $\mu:\mathbb{N}\to\mathbb{R}$ be such that univariate polynomials of degree $\leqslant n$ with coefficients in F can be multiplied with $O(\mu(n))$ arithmetic operations in F. Then we may take $\mu(n)=n\cdot\log n$ if F supports Fast Fourier Transforms and $\mu(n)=n\cdot\log n\cdot\log\log n$ for arbitrary F. See, e.g., [1, Section 7.4] or [6, Sections 2.1, 2.2] for algorithms, complexity results, and further references.

Polynomial division is specified as follows: For given $f,g\in F[z]$ with $\deg f=n\geqslant m=\deg g$, we want to compute the quotient q and remainder $r\in F[z]$ determined (uniquely) by $f = q \cdot g + r$ and $\deg r < m$. With the usual school algorithm, q and r can be computed with $O(m \cdot n)$ operations in F. For asymptotically fast algorithms, the bound $O(\mu(n-m)+\mu(m))$ is stated in [6, Corollary (2.26)]. With the school method for "blocks" of $O(m)$ coefficients, which are then processed with asymptotically fast techniques, polynomial division can be performed with $O((n/m) \cdot \mu(m))$ operations.

The greatest common divisor d of two polynomials $f, g \in F[z]$ with $\deg f \leqslant n$ and $\deg g \leqslant n$ can be computed with $O(\mu(n) \cdot \log n)$ arithmetic operations. Corresponding cofactors, i.e., polynomials $u, v \in F[z]$ with $u \cdot f + v \cdot g = d$ and $\deg(d \cdot u) < \deg g$ and $\deg(d \cdot v) < \deg f$, can be computed within the same time bound. This follows from [6, Corollary (3.14)].

Polynomial division and GCS computation are discussed further, e.g., in [1, Sections 8.3, 8.8]. The history of algorithms for these problems and further references are given, e.g., in [6, Sections 2.8, 3.8].

The algorithm for the Sylvester equation with Frobenius (companion) matrices A and B is based on the fact that if A is a Frobenius matrix and q is a polynomial, then the equation $q(A) \cdot y = d$ can be written in terms of polynomial arithmetic. The crucial result is Corollary 2.2 from [10]:

Lemma 4.1. *Let $A \in F^{m \times m}$ be a Frobenius matrix. Let $q \in F[z]$, and let $y = (y_0, y_1, \dots, y_{m-1})^{\mathrm{T}}$ and $d = (d_0, d_1, \dots, d_{m-1})^{\mathrm{T}} \in F^m$. Let $p_y(z) = y_0 + y_1 z + \cdots + y_{m-1} z^{m-1}$ and $p_d(z) = d_0 + d_1 z + \cdots + d_{m-1} z^{m-1}$. Then $q(A) \cdot y = d$ if and only if $q \cdot p_y \equiv p_d \bmod \chi_A$.*

Lemma 4.2. *Assume that in Lemma 3.6 the matrix $B \in F^{n \times n}$ is a Frobenius matrix. Then the r.h.s. d in Lemma 3.6 is $d = \sum_{k=0}^{n-1} \cdot A^k \cdot c_{k+1}$.*

Proof. If B is a Frobenius matrix, then $\chi_{B_k}(z) = z^k$ for $0 \leqslant k \leqslant n-1$. □

Lemma 4.3. *Let $A \in F^{m \times m}$ be a Frobenius matrix. Then d as in Lemma 4.2 can be computed with $O(m \cdot n)$ arithmetic operations.*

Proof. Multiplication of a vector with A can be done with $2m-1$ operations. The vector $d = \sum_{k=0}^{n-1} A^k \cdot c_{k+1}$ can be computed via Horner's rule with $n-1$ multiplications of A with a vector and $n-1$ additions in F^m. □

Lemma 4.4. *Let $A \in F^{m \times m}$ and $B \in F^{n \times n}$ be Frobenius matrices with disjoint spectra and $C \in F^{m \times n}$. Then the last column x_n of the solution X of $AX - XB^{\mathrm{T}} = C$ can be computed with $O(m \cdot n + \mu(m) \cdot \log m)$ arithmetic operations.*

Proof. According to Lemma 3.6, the vector x_n is the solution of the linear equation $\chi_B(A) \cdot x_n = d$, where the r.h.s. $d \in F^m$ is as in Lemma 4.2. Because of Lemma 4.1, this equation for x_n is equivalent to the polynomial relation $\chi_B \cdot p_x \equiv p_d \bmod \chi_A$, where p_x and p_d are the polynomials with coefficient vectors x_n and d, respectively. The characteristic polynomials χ_A and χ_B are relatively prime, because A and B have disjoint spectra. If u is a polynomial with $u \cdot \chi_B \equiv 1 \bmod \chi_A$, then p_x can be computed from the congruence $p_x \equiv u \cdot p_d \bmod \chi_A$, because then $\chi_B \cdot p_x \equiv \chi_B \cdot u \cdot p_d \equiv p_d \bmod \chi_A$.

The coefficients of χ_A and χ_B are given for free, because A and B are Frobenius (companion) matrices. The r.h.s. p_d can be computed with $O(m \cdot n)$ operations because of Lemma 4.3. The representative r of $\chi_B \bmod \chi_A$ of degree $< m$ can be computed with $O(m \cdot n)$ operations (school method for polynomial division), and then u with $\deg u < m$ and $u \cdot r \equiv u \cdot \chi_B \equiv 1 \bmod \chi_A$ can be computed with $O(\mu(m) \cdot \log m)$ operations with an extended GCD algorithm. Finally, p_x can be computed from u and p_d with $O(\mu(m))$ operations. □

Theorem 4.5. *Let $A \in F^{m\times m}$ and $B \in F^{n\times n}$ be Frobenius matrices with disjoint spectra and $C \in F^{m\times n}$. Then the solution X of $AX - XB^{\mathrm{T}} = C$ can be computed with $O(m \cdot n)$ arithmetic operations.*

Proof. We may assume $m \leqslant n$ w.l.o.g. (otherwise transpose the equation). Then $\mu(m) \cdot \log m = O(m^2) = O(m \cdot n)$. Hence the last column of X can be computed within the asserted time bound because of Lemma 4.4. As B is a Frobenius matrix, the other columns of X are given recursively by $x_{k-1} = Ax_k - b_{k,n}x_n - c_k$ for $n \geqslant k \geqslant 2$, cf. Lemma 3.7. Each step of this recursion can be done with $O(m)$ operations, hence $x_{n-1}, \ldots, x_1$ can be computed with $O(m \cdot n)$ operations. □

5. The generic case: reduction to Frobenius form

For the case of generic matrices A and B (i.e., the entries of A and B are indeterminates), the problem can be reduced to the Frobenius case by similarity transforms, using Keller-Gehrig's algorithm for computing the characteristic polynomial. The algorithm exploits fast matrix multiplication. Reduction to Frobenius Form is based on the following lemma.

Lemma 5.1. *Let $A \in F^{n\times n}$ and $v \in F^n \backslash \{0\}$. If the matrix $U = U(A) = (v, Av, A^2v, A^3v, \ldots, A^{n-1}v)$ is nonsingular, then $U^{-1}AU$ is a Frobenius matrix. If the entries of A are algebraically independent over a subfield F_0 of F and $v \in F_0^n \backslash \{0\}$, then $U(A)$ is nonsingular.*

Proof. Assume that U is nonsingular. Let e_j denote the jth unit vector. Then $U^{-1}AUe_j = U^{-1}AA^{j-1}v = U^{-1}A^jv = e_{j+1}$ for $1 \leqslant j < n$. So $U^{-1}AU$ is a Frobenius matrix.

Now let A have algebraically independent entries and $v \in F_0^n$. Choose $v_2, \ldots, v_n \in F_0^n$ such that $v = v_1, v_2, \ldots, v_n$ are a basis of F_0^n. Let $B \in F_0^{n\times n}$ be such that $Bv_i = v_{i+1}$ for $1 \leqslant i < n$. Then substitute B for the indeterminate matrix A in $U(A)$. This produces $U(B) = (v_1, v_2, \ldots, v_n)$, which is nonsingular. Therefore $\det U(B) \neq 0$ and hence $\det U(A) \neq 0$. □

A simple algorithm described by Keller-Gehrig [20, Section 3] shows that $U(A)$ can be computed with $O(n^\beta \cdot \log n)$ operations:

Lemma 5.2. *If the entries of A are algebraically independent, then the matrix U in Lemma* 5.1 *can be computed with* $\mathrm{O}(n^\beta \cdot \log n)$ *operations.*

The inverse of a nonsingular $n \times n$ matrix can be computed with $\mathrm{O}(n^\beta)$ operations, see [26] or [6, Section 16.4]. Now we have all ingredients to prove a time bound for the Sylvester equation in the generic case.

Proof of Theorem 1.1 (*Generic case*). Due to Lemma 5.2, matrices $U \in \mathrm{GL}(m)$ and $V \in \mathrm{GL}(n)$ such that $A' = U^{-1}AU$ and $B' = V^{-1}BV$ are Frobenius matrices can be computed with $\mathrm{O}(N^\beta \cdot \log N)$ operations. The matrix $C' = U^{-1}C(V^{-1})^{\mathrm{T}}$ can be computed with $\mathrm{O}(N^\beta)$ operations. The solution Y of the equation $A'Y - Y(B')^{\mathrm{T}} = C'$ can be computed with $\mathrm{O}(m \cdot n) \leqslant \mathrm{O}(N^\beta)$ operations, and the solution $X = UYV^{\mathrm{T}}$ of $AX - XB^{\mathrm{T}} = C$ can then be computed from Y with $\mathrm{O}(N^\beta)$ operations. □

6. The general case: divide and conquer

In general, a (square) matrix need not be similar to a Frobenius matrix. However, every matrix is similar to a block diagonal matrix, where the block diagonal entries are Frobenius matrices, e.g., its *first rational canonical form* or *Frobenius canonical form*, where the Frobenius blocks correspond to the invariant factors i_k of A, which are defined as follows: For each eigenvalue λ of A, let $\sigma_1(\lambda) \geqslant \sigma_2(\lambda) \geqslant \cdots \geqslant \sigma_n(\lambda) \geqslant 0$ denote the sizes of the Jordan blocks corresponding to λ in the Jordan canonical form of F. Let $\lambda_1, \ldots, \lambda_l$ denote the distinct eigenvalues of A. Then $i_k(z) = \prod_{j=1}^{l} (z - \lambda_j)^{\sigma_k(\lambda_j)}$ is the kth invariant factor of A. In particular, i_1 is the minimal polynomial of A, i_2 is the minimal polynomial of what remains when a largest Jordan block for each eigenvalue is removed, etc. The invariant factors are in $F[z]$, i_{k+1} divides i_k for each k, and $i_1, \ldots, i_n = \chi_A$. Another rational canonical form is the *second rational canonical form*, where the Frobenius blocks correspond to the elementary divisors of A, i.e., the characteristic polynomials of the Jordan blocks. For further information and proofs see [11, Sections 7.1–7.5] or [17, Section 3.4].

Unfortunately, the known algorithms for computing such canonical forms are not suited for the complexity bound of Theorem 1.1: Giesbrecht's $\mathrm{O}(N^\beta \cdot \log N)$ algorithm [12] is nondeterministic. The fastest known deterministic algorithm for the Frobenius canonical form (Storjohann, [25]) has no better complexity bound than $\mathrm{O}(N^3)$ and does not provide a transformation matrix. The known algorithms that compute a transformation matrix as well do not yield better time bounds than $\mathrm{O}(N^4)$, see [25] for further references.

Keller-Gehrig's algorithm for computing the characteristic polynomial in the general case transforms the matrix into a form that is more complicated and shows less information about the structure of the operator, but suffices for our purpose. The following definition is motivated by the fact that the aforementioned canonical forms correspond to the decomposition of the space into cyclic subspaces.

Definition 6.1. We call a matrix $A \in F^{n\times n}$ semicyclic if it is upper block triangular,

$$A = \begin{pmatrix} F_1 & * & * & \cdots & * \\ 0 & F_2 & * & & * \\ 0 & 0 & F_3 & & * \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & F_l \end{pmatrix}$$

with Frobenius matrices $F_1, \dots, F_l$ on the block diagonal.

The asterisks $*$ denote arbitrary entries. The following lemma is a direct consequence of this definition:

Lemma 6.2. *Any semicyclic matrix $A \in F^{n\times n}$ can be partitioned as*

$$\begin{array}{c} n_1 \\ n_0 \\ n_2 \end{array} \left(\begin{array}{c|c|c} A_1 & * & * \\ \hline 0 & A_0 & * \\ \hline 0 & 0 & A_2 \end{array}\right),$$

where $A_1 \in F^{n_1\times n_1}$ and $A_2 \in F^{n_2\times n_2}$ are semicyclic, $A_0 \in F^{n_0\times n_0}$ is a Frobenius matrix, $n_1, n_2, n_0, \geqslant 0$, $n = n_1 + n_2 + n_0$, and $n_1, n_2 \leqslant n/2$.

The dimensions n_1, n_2, n_0 are allowed to be zero. This simplifies the description of special cases. If, e.g., A itself is a Frobenius matrix, then we choose $n_1 = n_2 = 0$ and $A_0 = A$. The spectra of A_1, A_2, and A_0 are contained in the spectrum of A.

The crucial step in Keller-Gehrig's algorithm for computing the characteristic polynomial of a matrix A (not necessarily generic) is to transform A into a semicyclic matrix. An algorithm for this task is described in [20, Section 5] and [6, Section 16.6]. It yields the following time bound:

Lemma 6.3. *For $A \in F^{n\times n}$, a nonsingular matrix U such that $U^{-1}AU$ is semicyclic can be computed with $\mathrm{O}(n^\beta \cdot \log n)$ arithmetic operations.*

The divide and conquer algorithm for the Sylvester equation uses the following simple complexity result for rectangular matrix multiplication:

Lemma 6.4. *For $m \leqslant n$, the product of an $m \times m$ matrix A with an $m \times n$ matrix B can be computed with $\mathrm{O}(m^{\beta-1} \cdot n)$ operations.*

Proof. Partition $B = (B_1|B_2|\cdots|B_l)$ into $l = \lceil n/m \rceil$ blocks of size $m \times m$ and compute the products AB_j with $\mathrm{O}(m^\beta)$ operations each. This yields the overall complexity bound $\mathrm{O}(m^\beta \cdot l) = \mathrm{O}(m^{\beta-1} \cdot n)$. □

In the remainder of this section, we discuss the complexity of solving the Sylvester equation $AX - XB^{\mathrm{T}} = C$ for $A \in F^{m\times m}$ and $B \in F^{n\times n}$ with disjoint spectra.

Lemma 6.5. *Let A be semicyclic, B be a Frobenius matrix, and $m \leqslant n$. Then X can be computed with* $\mathrm{O}(m^{\beta-1} \cdot n)$ *arithmetic operations.*

Proof. If $m = 1$, then A is a Frobenius matrix. If A is a Frobenius matrix, then X can be computed with $\mathrm{O}(m \cdot n) = \mathrm{O}(m^{\beta-1} \cdot n)$ operations because of Theorem 4.5. If A is not a Frobenius matrix, then we partition A according to Lemma 6.2 and partition the rows of X and C accordingly. Then the Sylvester equation reads

$$\begin{pmatrix} A_1 & U_1 & U_2 \\ 0 & A_0 & U_3 \\ 0 & 0 & A_2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_0 \\ X_2 \end{pmatrix} - \begin{pmatrix} X_1 \\ X_0 \\ X_2 \end{pmatrix} B^{\mathrm{T}} = \begin{pmatrix} C_1 \\ C_0 \\ C_2 \end{pmatrix}$$

or equivalently

$$A_2 X_2 - X_2 B^{\mathrm{T}} = C_2, \tag{6.1}$$

$$A_0 X_0 - X_0 B^{\mathrm{T}} = C_0 - U_3 X_2, \tag{6.2}$$

$$A_1 X_1 - X_1 B^{\mathrm{T}} = C_1 - U_1 X_0 - U_2 X_2. \tag{6.3}$$

Let $T(M,N)$ denote a time bound for solving the Sylvester equation for the special case specified in the lemma with $m \leqslant M$ and $n \leqslant N$. Then X_2 can be computed from (6.1) in time $T(M/2,N)$. The r.h.s. of (6.2) can be computed in time $\mathrm{O}(M^{\beta-1} \cdot N)$ because of Lemma 6.4. The solution X_0 of (6.2) can be computed with $\mathrm{O}(M \cdot N)$ operations because of Theorem 4.5. The r.h.s. of (6.3) can also be computed in time $\mathrm{O}(M^{\beta-1} \cdot N)$, and X_1 can then be computed from (6.3) in time $T(M/2,N)$. This shows that $T(M,N)$ fulfils the recursive estimate

$$T(M,N) \leqslant 2 \cdot T(M/2,N) + \mathrm{O}(M^{\beta-1} \cdot N), \qquad T(1,N) \leqslant \mathrm{O}(N),$$

which implies the assertion of the lemma. □

Lemma 6.6. *Let A be semicyclic, B be a Frobenius matrix, and $m \geqslant n$. Then X can be computed with* $\mathrm{O}(m^{\beta})$ *arithmetic operations.*

Proof. Let $T(M,N)$ denote a time bound for the case discussed here with $m \leqslant M$ and $n \leqslant N$. We reduce to smaller cases by partitioning A as in the proof of Lemma 6.5. The time for computing the r.h.s. of (6.2) and (6.3) is bounded by $\mathrm{O}(M^{\beta})$. (We do not exploit the fact that the matrix multiplications involved here can be computed cheaper if n is small compared with m, see Section 7.) The recursive estimate is as follows:

$$T(M,N) \leqslant 2 \cdot T(M/2,N) + \mathrm{O}(M^{\beta}), \qquad T(N,N) \leqslant \mathrm{O}(N^{\beta}).$$

This implies $T(M,N) = \mathrm{O}(M^{\beta})$. □

We summarize Lemmas 6.5 and 6.6. The role of A and B may be exchanged by transposing the equation.

Corollary 6.7. *Let A, B be semicyclic, one of them a Frobenius matrix, and $m, n \leqslant N$. Then X can be computed with $\mathrm{O}(N^{\beta})$ arithmetic operations.*

Lemma 6.8. *Let A, B be semicyclic and $m, n \leqslant N$. Then X can be computed with $\mathrm{O}(N^{\beta})$ arithmetic operations.*

Proof. We partition both A and B according to Lemma 6.2. The Sylvester equation has the form

$$\begin{pmatrix} A_1 & U_1 & U_2 \\ 0 & A_0 & U_3 \\ 0 & 0 & A_2 \end{pmatrix} X - X \begin{pmatrix} B_1^{\mathrm{T}} & 0 & 0 \\ V_1^{\mathrm{T}} & B_0^{\mathrm{T}} & 0 \\ V_2^{\mathrm{T}} & V_3^{\mathrm{T}} & B_2^{\mathrm{T}} \end{pmatrix} = C.$$

The matrices C and X are partitioned accordingly,

$$X = \begin{pmatrix} X_{1,1} & X_{1,0} & X_{1,2} \\ X_{0,1} & X_{0,0} & X_{0,2} \\ X_{2,1} & X_{2,0} & X_{2,2} \end{pmatrix}, \quad C = \begin{pmatrix} C_{1,1} & C_{1,0} & C_{1,2} \\ C_{0,1} & C_{0,0} & C_{0,2} \\ C_{2,1} & C_{2,0} & C_{2,2} \end{pmatrix}.$$

This implies nine smaller Sylvester equations for the $X_{i,j}$ $(i, j \in \{1, 0, 2\})$, in which the underlined matrix can be computed from the ones computed previously:

$$A_2\underline{X_{2,2}} - \underline{X_{2,2}}B_2^{\mathrm{T}} = C_{2,2},$$

$$A_2\underline{X_{2,0}} - \underline{X_{2,0}}B_0^{\mathrm{T}} - X_{2,2}V_3^{\mathrm{T}} = C_{2,0},$$

$$A_2\underline{X_{2,1}} - \underline{X_{2,1}}B_1^{\mathrm{T}} - X_{2,0}V_1^{\mathrm{T}} - X_{2,2}V_2^{\mathrm{T}} = C_{2,1},$$

$$A_0\underline{X_{0,2}} + U_3X_{2,2} - \underline{X_{0,2}}B_2^{\mathrm{T}} = C_{0,2},$$

$$A_0\underline{X_{0,0}} + U_3X_{2,0} - \underline{X_{0,0}}B_0^{\mathrm{T}} - X_{0,2}V_3^{\mathrm{T}} = C_{0,0},$$

$$A_0\underline{X_{0,1}} + U_3X_{2,1} - \underline{X_{0,1}}B_1^{\mathrm{T}} - X_{0,0}V_1^{\mathrm{T}} - X_{0,2}V_2^{\mathrm{T}} = C_{0,1},$$

$$A_1\underline{X_{1,2}} + U_1X_{0,2} + U_2X_{2,2} - \underline{X_{1,2}}B_2^{\mathrm{T}} = C_{1,2},$$

$$A_1\underline{X_{1,0}} + U_1X_{0,0} + U_2X_{2,0} - \underline{X_{1,0}}B_0^{\mathrm{T}} - X_{1,2}V_3^{\mathrm{T}} = C_{1,0},$$

$$A_1\underline{X_{1,1}} + U_1X_{0,1} + U_2X_{2,1} - \underline{X_{1,1}}B_1^{\mathrm{T}} - X_{1,0}V_1^{\mathrm{T}} - X_{1,2}V_2^{\mathrm{T}} = C_{1,1}.$$

The r.h.s. of all these equations can be computed with $\mathrm{O}(N^{\beta})$ operations. Five of the equations involve one of the Frobenius matrices A_0 or B_0 as a coefficient matrix. These equations can be solved with $\mathrm{O}(N^{\beta})$ operations because of Corollary 6.7. The remaining four equations are solved recursively. The time bound $T(N)$ is estimated recursively by

$$T(N) \leqslant 4 \cdot T(N/2) + \mathrm{O}(N^{\beta}),$$

which implies $T(N) \leqslant \mathrm{O}(N^{\beta})$. □

Proof of Theorem 1.1 (*General case*). Compute nonsingular matrices $U \in F^{m\times m}$ and $V \in F^{n\times n}$ such that $A' = U^{-1}AU$ and $B' = V^{-1}BV$ are semicyclic. U and V can be computed with $\mathrm{O}(N^{\beta} \cdot \log N)$ operations (Lemma 6.3), and A' and B' can then be computed with $\mathrm{O}(N^{\beta})$ operations. The matrices A' and B' have disjoint spectra, and the (unique) solution $Y \in F^{m\times n}$ of $A'Y - Y(B')^{\mathrm{T}} = U^{-1}C(V^{-1})^{\mathrm{T}}$ can be computed according to Lemma 6.8 with $\mathrm{O}(N^{\beta})$ operations. The solution $X = UYV^{\mathrm{T}}$ of the original equation $AX - XB^{\mathrm{T}} = C$ can now be computed with $\mathrm{O}(N^{\beta})$ operations. □

7. Conclusion

We have shown that if the Sylvester equation has a unique solution, then this solution can be computed with $\mathrm{O}(N^{\beta} \cdot \log N)$ rational operations, where we can choose $\beta < 2.376$. The present algorithm is the first rational algorithm that is competitive (in terms of arithmetic operations) with and even faster than the classical algorithms from numerical linear algebra. This is aesthetically pleasing and useful for generalizations for other fields than $\mathbb{R}$ or $\mathbb{C}$.

Before discussing major open questions, we wish to point out that it should be checked out whether even better time bounds for solving the Sylvester equation for $m \ll n$ can be obtained via *rectangular matrix multiplication*. Namely, the product of an $N \times N$ matrix with an $N \times N^{\alpha}$ matrix can be computed with $\mathrm{O}(N^{2+\varepsilon})$ operations for any $\varepsilon > 0$, if $\alpha < 0.294$ [7]. Generations of this result are given in [19].

Two major questions are still open: first, can these ideas be used to construct a numerically stable (and practically useful, at least for $\beta = 2.81$) algorithm, following the lines of, e.g., [16, Chapter 22, 15, 4], and second, what about the case where the spectra of A and B are not disjoint? While we do not have an answer for the latter problem, we can say a little bit about numerical stability.

Many people consider Keller-Gehrig's algorithm to be numerically unstable. Nevertheless, it provides a stable way to compute the characteristic polynomial and, in combination with a fast zero finding method, the eigenvalues of a complex matrix. This leads to favourable bit complexity bounds for the problem of computing eigenvalues [23, Section 21].

The problem with Keller-Gehrig's algorithm is that the transformation matrix U of Lemma 5.2 resp. Lemma 6.3, which transforms a matrix A to semicyclic form, may be ill-conditioned. It is possible to control this condition problem and derive a numerical version of our algorithm for the Sylvester equation which is numerically stable in a restricted sense. The analysis of this algorithm yields that under reasonable normalizing conditions a matrix X with $\|AX - XB^{\mathrm{T}} - C\| \leqslant 2^{-s}$ can be computed with $\mathrm{O}(N^{\beta+o(1)} \cdot \psi(N \cdot s))$ bit operations, where $\psi(L)$ is a time bound for L bit integer multiplication, e.g., $\psi(L) = \mathrm{O}(L \cdot \log L \cdot \log\log L)$. A precise specification and details can be found in [21].

Acknowledgements

I am grateful for helpful comments from and discussions with T. Ahrendt, M. Bläser, D. Lauer, and A. Schönhage.

References

[1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, D. Sorensen, LAPACK Users' Guide, 2nd Edition, SIAM, Philadelphia, PA, 1995.

[3] W.F. Arnold, A.J. Laub, Generalized eigenproblem algorithms and software for algebraic Riccati equations, Proc. IEEE 72 (1984) 1746–1754.

[4] D.H. Bailey, H.R.P. Ferguson, A Strassen–Newton algorithm for high-speed parallelizable matrix inversion, Proc. Supercomputing '88: November 14–18, Orlando, FL, vol. 1, IEEE Computer Society Press, Silver Spring, MD, USA, 1988, pp. 419–424.

[5] R.H. Bartels, G.W. Stewart, Algorithm 432: solution of the matrix equation $AX + XB = C$, Comm. ACM 15 (9) (1972) 820–826.

[6] P. Bürgisser, M. Clausen, M.A. Shokrollahi, Algebraic Complexity Theory, Springer, Berlin, 1997.

[7] D. Coppersmith, Rectangular matrix multiplication revisited, J. Complexity 13 (1) (1997) 42–49.

[8] D. Coppersmith, S. Winograd, Matrix multiplication via arithmetic progressions, J. Symbolic Comput. 9 (1990) 251–280.

[9] J. Demmel, Three methods for refining estimates of invariant subspaces, Computing 38 (1987) 43–57.

[10] C.M. Fiduccia, An efficient formula for linear recurrences, SIAM J. Comput. 14 (1) (1985) 106–112.

[11] F.R. Gantmacher, Matrizentheorie, Springer, Berlin, 1986.

[12] M. Giesbrecht, Nearly optimal algorithms for canonical matrix forms, SIAM J. Comput. 24 (5) (1995) 948–969.

[13] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd Edition, John Hopkins Studies in the Mathematical Sciences, The Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[14] G.H. Golub, S. Nash, C. Van Loan, A Hessenberg–Schur method for the matrix problem $AX + XB = C$, IEEE Trans. Automat. Control AC-24 (6) (1979) 909–913.

[15] N.J. Higham, Exploiting fast matrix multiplication within the level 3 BLAS, ACM Trans. Math. Software 16 (4) (1990) 352–368.

[16] N.J. Higham, Accuracy and Stability of Numerical Algorithms, SIAM, Philadelphia, PA, 1996.

[17] R.A, Horn, C.A. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.

[18] R.A, Horn, C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, Cambridge, 1994.

[19] X. Huang, V.Ya. Pan, Fast rectangular matrix multiplication and applications, J. Complexity 14 (2) (1998) 257–299.

[20] W. Keller-Gehrig, Fast algorithms for the characteristic polynomial, Theoret. Comput. Sci. 36 (1985) 309–317.

[21] P. Kirrinnis, Fast computation of invariant subspaces and bit complexity in numerical linear algebra, Habilitationsschrift, University of Bonn, Department Computer Science, Bonn, November 1999.

[22] A.J. Laub, Invariant subspace methods for the numerical solution of Riccati equations, Chapter 7, in: A.J. Laub, S. Bittanti, J.C. Willems (Eds.), Ricatti Equations, Springer, Berlin, June 4–6 1990, pp. 163–196.

[23] A. Schönhage, The fundamental theorem of algebra in terms of computational complexity, Technical Report, University of Tübingen, 1982.

[24] G. Starke, W. Niethammer, SOR for $AX - XB = C$, Linear Algebra Appl. 154/156 (1991) 355–375.

[25] A. Storjohann, An $O(n^3)$ algorithm for the Frobenius normal form, in: O. Gloor (Ed.), ISSAC 98: Proc. 1998 Internat. Symp. on Symbolic and Algebraic Computation, August 13–15, 1998, University of Rostock, Germany, ACM Press, New York, 1998, pp. 101–105.

[26] V. Strassen, Gaussian elimination is not optimal, Numer. Math. 13 (1969) 354–356.

[27] S. Winograd, On multiplication of 2 × 2 matrices, Linear Algebra Appl. 4 (1971) 381–388.