Triangular x-basis decompositions and derandomization of linear algebra algorithms over $\mathsf{K}[x]$

Somit Gupta, Soumojit Sarkar, Arne Storjohann [*]

Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Johnny Valeriote

Centre for Computational Mathematics in Industry and Commerce, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Abstract

Deterministic algorithms are given for some computational problems that take as input a nonsingular polynomial matrix A over $\mathsf{K}[x]$, K an abstract field, including solving a linear system involving A and computing a row reduced form of A. The fastest known algorithms for linear system solving based on the technique of high-order lifting by Storjohann (2003), and for row reduction based on fast minimal approximant basis computation algorithm by Giorgi, Jeannerod and Villard (2003), use randomization to find either a linear or small degree polynomial that is relatively prime to $\det A$. We derandomize these algorithms by first computing a factorization of $A = UH$, with x not dividing $\det U$ and $x - 1$ not dividing $\det H$. A partial linearization technique, that is applicable also to other problems, is developed to transform a system involving H, which may have some columns of large degrees, to an equivalent system that has degrees reduced to that of the average column degree.

Key words: polynomial matrices, linear system solving, row reduction, derandomization

1. Introduction

Let K be a field and x be an indeterminant. This paper considers algorithms for linear algebra problems over $\mathsf{K}[x]$, the ring of univariate polynomials over K. Let $A \in \mathsf{K}[x]^{n\times n}$

[*] Corresponding author.

Email addresses: somit.gupta@gmail.com (Somit Gupta), soumojitsarkar@gmail.com (Soumojit Sarkar), astorjoh@uwaterloo.ca (Arne Storjohann), johnnyvaleriote@gmail.com (Johnny Valeriote).

be an input matrix. Problems involving A include computing the rank, a nullspace, the determinant and Smith form, and solving a linear system. These problems have received a lot of attention, with the main goal recently being to reduce the cost to about the same (in terms of field operations from K) as that of multiplying together two input matrices with the same dimension and degree of entries as A, that is, within the cost bound of $O\tilde{~}(n^{\omega}d)$ field operations from K, where d is a bound on the degrees of entries in A and ω is the exponent of matrix multiplication. For surveys on this topic we refer to (Storjohann, 2003; Giorgi et al., 2003; Jeannerod and Villard, 2005). All of the problems mentioned above have Las Vegas randomized algorithms which support the target cost $O\tilde{~}(n^{\omega}d)$, but to the best of our knowledge fully deterministic algorithms with this cost are not known. In this paper we give deterministic reductions to matrix multiplication for two of the problems on a nonsingular $A \in \mathsf{K}[x]^{n\times n}$: linear system solving and row reduction. We now discuss each of these problems in more detail.

Linear system solving takes as input a nonsingular $A \in \mathsf{K}[x]^{n\times n}$, together with a vector $b \in \mathsf{K}[x]^{n\times 1}$, and asks as output the unique vector $v := A^{-1}b \in \mathsf{K}(x)^{n\times 1}$. The high-order lifting technique of Storjohann (2003) gives a reduction of linear system solving to matrix multiplication. High-order lifting requires an $X \in \mathsf{K}[x]$ of small degree that is relatively prime to $\det A$ (denoted by $X \perp \det A$), and computes the X-adic series expansion of $A^{-1}b$ to high enough precision to allow the solution vector to be recovered using rational function reconstruction. Once a suitable X is known the rest of the algorithm is deterministic. The ideal choice for X from a practical point of view is $X = x^d$ since this allows working in the standard power basis. If x divides the determinant of A, current methods appeal to randomization. If the size of K is large enough, the input system (A, b) can be shifted with a change of variable $x \to x - \alpha$ for a random $\alpha \in \mathsf{K}$ such that x does not divide $\det A\mid_{x=x-\alpha}$ with high probability. If the size of K is too small, we can work over an algebraic extension of K of degree bounded by $O(\log nd)$ to afford sufficiently many choices for the random shift (incurring a multiplicative factor of $(\log nd)^{1+o(1)}$ in the cost), or choose X to be an irreducible of degree larger than one. In this paper we show how to avoid the need for randomization by developing an algorithm that always allows the choice $X = x^d$.

Our approach is to first decompose A as the product of two matrices: $A = UH$. Let us define $\deg A := \max_{i,j} \deg A_{ij}$. If $\deg A \leq d$, then the factor U produced by our algorithm will satisfy $\deg U \leq d$ also, while the matrix H will be upper triangular with powers of x on the diagonal, and offdiagonal entries of degree strictly less than the diagonal entry in the same column. Here is an example of a 3×3 matrix of degree 2 over $\mathbb{Z}/(7)[x]$.

$$\overset{A}{\begin{bmatrix} x^2 & x+1 & x+4 \\ x & x^2+5x & 6x+1 \\ 0 & 3x+5 & x^2+6x+6 \end{bmatrix}} = \overset{U}{\begin{bmatrix} x & x+1 & 1 \\ 1 & x^2+5x & 3x+5 \\ 0 & 3x+5 & 2 \end{bmatrix}} \overset{H}{\begin{bmatrix} x & 0 & 2x^2+1 \\ & 1 & 4x^2+3x+4 \\ & & x^3 \end{bmatrix}} \qquad (1)$$

The matrix H in (1) can be considered to be a local Hermite form of A at $x = 0$, similar to the local Smith form presented by Wilkening and Yu (2011). Our algorithm for computing U and H does not actually recover $\det U$, but the decomposition $\det A = (\det U)\cdot(\det H)$

does split the determinant of A into two parts: $\det H$ is a power of x, while x does not divide $\det U$. For the example in (1) we have

$$\det A = (\det U) \times (\det H) = (x^2 + 4x + 3) \times x^4.$$

We call $A = UH$ an *x-Hermite decomposition* of A. More generally, if we keep the same conditions on $\det U$ and $\det H$ but don't insist that H be in Hermite form, we call $A = UH$ an *x-basis decomposition.* Once an x-basis decomposition is known, X-adic lifting can be used to solve the system $Av = b$ for v in two steps: first compute $u := U^{-1}b$ followed by $v := H^{-1}u$, using X a power of x for U, and X a power of $x - 1$ for H. (Every field K, even an abstract field, contains the two linear irreducibles x and $x - 1$.) We give two algorithms for computing an x-basis decomposition. The first algorithm is based on the technique of linear x^d-adic lifting (see Dixon, 1982; Moenck and Carter, 1979) and computes the canonical x-Hermite decomposition as shown in (1). The algorithm runs in time $O(n^3 \,\mathsf{M}(d))$ field operations, where $\mathsf{M}(d)$ bounds the cost of polynomial multiplication of degree d. While linear x^d-adic lifting costs $O\tilde{}(n^3 d)$ to solve a single linear system $A^{-1}b$ which has numerators and denominators bounded in degree by $O(nd)$, we use an amortized analysis to achieve an $O(n^3 \,\mathsf{M}(d))$ running time overall by exploiting the fact that the sum of the column degrees of H will be equal to $\deg \det A$, which is bounded by nd.

Our second algorithm for x-basis decomposition incorporates matrix multiplication. As shown in (1), some column degrees in H may be larger than others, even as large as nd where d is the degree of the input matrix A. In general, the individual column degrees of H can not be predicted well *a priori*. We solve this by conditioning the input matrix A by postmultiplying by a permutation P that ensures that each diagonal entry in the H corresponding to the x-Hermite decomposition of AP will divide the next. With this property in hand, we know that the first $n/2$ columns of H will be bounded in degree by $2d$, the next $n/4$ columns will be bounded in degree by $4d$, and so on. Our fast algorithm for x-basis decomposition iterates for $i = 1, 2, \ldots, O(\log n)$. At iteration i we work over both $\mathsf{K}[x]$ and its residue class ring $\mathsf{K}[x]/(x^{2^i d+1})$ to partially condition the matrix to allow recovery of the next $n/2^i$ columns. The computations over $\mathsf{K}[x]$ exploit a precision × dimension compromise. Roughly speaking, at each stage the column dimension is halved but the precision doubled. The computations over $\mathsf{K}[x]/(x^{2^i d+1})$ use an algorithm that recurses on the precision $t := 2^i d + 1$, reducing the problem of precision t to two subproblems of precision about $t/2$. The asymptotically fast version of the x-basis decomposition algorithm has cost $O(n^\omega (\log n)^2 \,\mathsf{M}(d))$ field operations from K, and is applicable over any field K. The algorithm produces the x-basis decomposition $A = U(HP^{-1})$ of A, corresponding to the x-Hermite decomposition $AP = UH$ of AP.

Producing an x-basis decomposition $A = UH$, either using the iterative or the fast algorithm, is not actually sufficient to achieve derandomization of the algorithms for our target problems, for example linear system solving via $u := U^{-1}b$ followed by $v := H^{-1}u$. The difficulty with using the x-basis decomposition computed by our algorithms is that H may have entries of degree $\Omega(nd)$. Techniques such as high-order lifting and integrality certification (Storjohann, 2003) to compute $H^{-1}u$ and $\det A$ are highly sensitive to the degree of the largest entry in the input matrix. However, we can observe that the sum $E = \sum_{j=1}^{n} \deg \mathrm{Col}(H, j)$ of the degrees of the columns of H will be equal to $\deg \det A$. Thus, the average column degree E/n of H is exactly $(\deg \det A)/n$, which is bounded by the degree d of the input matrix A. To solve the problem of some columns of H

having large degree, we prove that corresponding to H there always exists a matrix D of dimension strictly less than $2n$ that satisfies the following properties: $\deg D \leq \lceil E/n \rceil \leq d$ and H^{-1} is equal to the principal $n \times n$ submatrix of D^{-1}. The system solution $H^{-1}u$ can then be recovered as the first n components of the vector $D^{-1}\bar{u}$, where $\bar{u}$ is u augmented with some zero entries. The example just given was for partial column linearization. More generally, we give an approach for partially linearizing a matrix that has some large degree rows and/or columns. The technique is applicable to a wide variety of problems, such as rank, adjoint, determinant and Smith form computation. The transformation of a given input matrix such as H to its partially linearized form D does not require any computation in terms of field operations from K, and is effective over any field.

Now consider the row reduction problem. Row reduction produces a matrix R such that the set of all $\mathsf{K}[x]$-linear combinations of rows of R is equal to the set of all $\mathsf{K}[x]$-linear combinations of rows of A, but the rows of R have degrees as small as possible. Thus, row reduction is essentially lattice reduction for polynomial matrices. The fastest algorithm to compute a row reduced form (Giorgi et al., 2003) uses high-order lifting and fast minimal approximant basis computation. The first step of the algorithm is to randomly shift $x \to x - \alpha$ to ensure that $x \perp \det A$. Unlike the linear system solving problem, if K is too small, the algorithm of Giorgi et al. (2003) does not seem directly amenable to working modulo an irreducible X that is nonlinear. Working over an extension field is also problematic because entries in the resulting reduced form R may be over the extension and not the ground field. In this paper we show how to derandomize the algorithm for row reduction by first computing an x-basis decomposition $A = UH$, then applying our partial linearization technique to allow fast computation of a row reduced form R_1 of H, and finally using the approach of Giorgi et al. (2003) to compute a so called shifted row reduced form R_2 of the matrix AR_1^{-1}, which we can show will be over $\mathsf{K}[x]$ with degree bounded by d and with $x \perp \det AR_1^{-1}$, to arrive at a row reduced form R_2R_1 of A.

The rest of this paper is organized as follows. Section 2 defines our cost model and discusses the computation of ring operations over the polynomial ring $\mathsf{K}[x]$, as well as its residue class rings $\mathsf{K}[x]/(x^t)$; some facts about triangular and diagonal forms over $\mathsf{K}[x]/(x^t)$ are also recalled. Section 3 gives our $O(n^3\,\mathsf{M}(d))$ field operations algorithm for computing the canonical x-Hermite decomposition. Sections 4 and 5 give the $O(n^\omega (\log n)^2\,\mathsf{M}(d))$ algorithm for x-basis decomposition. Section 6 explains the partial linearization transformation; this section will be of independent interest. Section 7 applies the results of the previous sections to give a deterministic reduction of rational linear system solving to matrix multiplication. Section 8 gives the deterministic algorithm for row reduction, and Section 9 concludes.

2. Cost model and preliminaries

Algorithms are analysed by bounding the number of required field operations from a field K on an algebraic random access machine; the operations $+$, $-$, $\times$ and "divide by a nonzero" involving two field elements have unit cost.

We use ω to denote the exponent of matrix multiplication: two $n \times n$ matrices over a ring R can be multiplied with $O(n^\omega)$ ring operations from R. We use M for polynomial multiplication: let $\mathsf{M} : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{>0}$ be such that polynomials in $\mathsf{K}[x]$ of degree bounded by d can be multiplied using at most $\mathsf{M}(d)$ field operations from K. We refer to von zur Gathen and Gerhard (2003) for more details and references about ω and M. We assume that $2 < \omega \leq 3$, and that $\mathsf{M}(ab) \leq \mathsf{M}(a)\mathsf{M}(b)$ for $a, b \in \mathbb{Z}_{>1}$. Some of our complexity estimates will explicitly make the assumption that $\mathsf{M}(d) \in O(d^{\omega-1})$. This assumption on M states that if fast matrix multiplication techniques are used, then fast polynomial multiplication should also be used.

Given two polynomials $a, b \in \mathsf{K}[x]$ with b nonzero, we denote by $\mathrm{Rem}(a, b)$ and $\mathrm{Quo}(a, b)$ the unique polynomials such that $a = \mathrm{Quo}(a, b)\, b + \mathrm{Rem}(a, b)$, subject to the degree contraint $\deg \mathrm{Rem}(a, b) < \deg b$. If a and b have degree bounded by d then both the Rem and Quo operations have cost $O(\mathsf{M}(d))$, and if b is a power of x both operations are free in our cost model. If the first argument of Rem or Quo is a matrix or vector the intention is to apply the function elementwise to the entries.

Given a matrix $A \in \mathsf{K}[x]^{n\times n}$ of degree d that is nonsingular modulo x, together with a $B \in \mathsf{K}[x]^{n\times m}$, high-order lifting (Storjohann, 2003) can be used to compute the truncated x-adic expansion $\mathrm{Rem}(A^{-1}B, x^{sd+1})$ up to a desired order s. The cost depends on m, the column dimension of B, and s, the desired order. Storjohann (2003) describes an algorithm that exploits the case when B has small degree: $\deg B \leq d$. If $m = 1$ and $\deg B \leq d$, then the algorithm supporting (Storjohann, 2003, Proposition 15) computes $\mathrm{Rem}(A^{-1}B, x^{sd+1})$ in time $O(n^\omega(\log s + s/n)\, \mathsf{M}(d))$. Note that if $s \in O(n \log n)$ then this cost estimate simplifies to $O(n^\omega(\log n)\, \mathsf{M}(d))$. The algorithm is easily modified to accommodate the case when $m > 1$ without impacting the running time, provided that the precision × dimension invariant $s \times m \in O(n \log n)$ is satisfied. In particular, there are two phases of the algorithm that require computation. Phase 1 does not depend on m and has running time $O(n^\omega(\log s)\, \mathsf{M}(d))$, while the cost of phase 2 (a loop) is dominated by the last iteration which requires the multiplication of an $n \times n$ matrix of degree d with an $n \times sm$ matrix of degree d; if $s \times m \in O(n \log n)$ then this multiplication has cost $O(n^\omega(\log n)\, \mathsf{M}(d))$. The following result will be used in Subsection 5.3.

Theorem 1. *Let $A \in \mathsf{K}[x]^{n\times n}$ (with $\mathrm{Rem}(A, x)$ nonsingular) and $B \in \mathsf{K}[x]^{n\times m}$ both have degrees of entries bounded by d. If s satisfies $s \times m \in O(n \log n)$, then high-order lifting can be used to compute $\mathrm{Rem}(A^{-1}B, x^{sd+1})$ in $O(n^\omega(\log n)\, \mathsf{M}(d))$ field operations from K.*

The extended gcd problem takes as input two polynomials $a, b \in \mathsf{K}[x]$, and asks as output the polynomials $g, s, t, u, v \in \mathsf{K}[x]$ such that

$$\begin{bmatrix} s & t \\ u & v \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} g \\ \ \end{bmatrix}, \tag{2}$$

with g a greatest common divisor of a and b, and $sv - tu$ a nonzero constant polynomial. It will be useful to define an additional function B to bound the cost of the extended gcd operation, as well as other gcd-related computations. We can take $\mathsf{B}(d) = \mathsf{M}(d) \log d$ or $\mathsf{B}(d) = d^2$. Then the extended gcd problem with two polynomials in $\mathsf{K}[x]$ of degree bounded by d can be solved in time $O(\mathsf{B}(d))$.

Our algorithm in Section 5 for computing an x-basis decomposition over $\mathsf{K}[x]$ works by passing back and forth between the principal ideal domain $\mathsf{K}[x]$ and its residue class rings $\mathsf{K}[x]/(x^d)$ for various values of d. In the remainder of this section we briefly discuss the computation of ring operations, and recall some facts about the unimodular triangularization and diagonalization of matrices over $\mathsf{K}[x]/(x^d)$.

Computing over $\mathsf{R} = \mathsf{K}[x]/(x^d)$

We identify R with the set $\{a \in \mathsf{K}[x] \mid \deg a < d\}$. A homomorphism $\phi_d : \mathsf{K}[x] \to \mathsf{R}$ can be naturally defined as $\phi_d(a) = \mathrm{Rem}(a, x^d)$ for $a \in \mathsf{K}[x]$. The set of units in R is the set of elements with nonzero constant coefficient. Every nonzero element $a \in \mathsf{R}$ can be written uniquely as $a = \tilde{a}x^e$ where $\tilde{a}$ is a unit and $0 \leq e < d$ is called the trailing degree of a. The trailing degree of zero is $-\infty$. For any two elements in R, not both zero, the nonzero element with the smallest trailing degree divides the other, and all elements with the same trailing degree are associates of each other. The proscribed complete set of nonassociates of R is the set $\{0, x, x^2, \ldots, x^{d-1}\}$.

Addition and subtraction of two elements has cost $O(d)$, while multiplication has cost $O(\mathsf{M}(d))$. The extended gcd problem (see (2)) over R also has cost $O(\mathsf{M}(d))$. If either of a or b is zero then either $(s, t, u, v) := (1, 0, 0, 1)$ or $(s, t, u, v) := (0, 1, 1, 0)$ will work. Now assume both a and b are nonzero. Up to swapping a and b, if required, we can assume without loss of generality that a is a gcd of a and b: write $a = \tilde{a}x^e$ and $b = \tilde{b}x^f$ with $\tilde{a}$ and $\tilde{b}$ both units from R, and with $e \leq f$. Set $(s, t, u, v) := (1, 0, -\tilde{b}\tilde{a}^{-1}x^{f-e}, 1)$. Newton iteration (see von zur Gathen and Gerhard, 2003, Algorithm 9.3) can be used to compute $\tilde{a}^{-1}$ in time $O(\mathsf{M}(d))$.

Triangular forms over $\mathsf{R} = \mathsf{K}[x]/(x^d)$

The unimodular matrices over R are precisely those with determinant a unit. Corresponding to every matrix $A \in \mathsf{R}^{n\times m}$ are unimodular matrices $U \in \mathsf{R}^{n\times n}$ and $V \in \mathsf{R}^{m\times m}$ such that UAV is in Smith canonical form: $S = UAV$ is zero except for the diagonal entries which are coming from the proscribed complete set of nonassociates of R, namely $S_{11}, S_{22}, \ldots, S_{rr}, 0, \ldots, 0 = x^{e_1}, x^{e_2}, \ldots, x^{e_r}, 0, \ldots, 0$, with $0 \leq e_1 \leq \cdots \leq e_r$, $r \leq \min(n, m)$. The Smith form over R always exists and is unique (Kaplansky, 1949, Theorem 9.3).

An $n\times m$ matrix over $\mathsf{K}[x]$ or $\mathsf{R} = \mathsf{K}[x]/(x^d)$ is in row Hermite form if it is in row echelon form with pivot entries nonzero elements of the proscribed complete set of nonassociates, and offdiagonal entries in pivot columns of degree less than the pivot entry in the same column. While the Hermite form is a canonical form over the principal ideal domain $\mathsf{K}[x]$, it is not over R, a principal ideal ring with zero divisors. The following example is over $\mathsf{K}[x]/(x^3)$.

$$\overset{U}{\begin{bmatrix} 1 & \\ x & 1 \end{bmatrix}} \begin{bmatrix} x^2 & x \\ & \end{bmatrix} = \begin{bmatrix} x^2 & x \\ & x^2 \end{bmatrix}. \tag{3}$$

A canonical form for left equivalence over R is given by the Howell form as described by Howell (1986) (also see Storjohann and Mulders, 1998). A matrix H is in Howell form if it is in Hermite form and satisfies the following additional condition: for any j, $0 \leq j \leq m$, if $v \in \mathsf{R}^{1\times m}$ has first j entries zero and is an R-linear combination of the rows of H, then v is an R-linear combination of the subset of rows of H that have first j entries zero. For example, the matrix on the right of (3) is in Howell form over $\mathsf{K}[x]/(x^3)$ while the matrix on the left is not. The Howell form is a canonical form for left equivalence that has a maximal number of nonzero rows among any echelon form.

Although the Hermite form is not a canonical form in general over R, some matrices enjoy the property of having a unique Hermite form.

Definition 2. A matrix $H \in \mathsf{R}^{n\times m}$ is said to be in triangular Smith form if it is in Hermite form, and the nonzero rows of H can be written as

$$\begin{bmatrix} x^{e_1} & h_{12} & h_{13} & \cdots & h_{1r} & \cdots & h_{1m} \\ & x^{e_2} & h_{23} & \cdots & h_{2r} & \cdots & h_{2m} \\ & & x^{e_3} & & \vdots & & \vdots \\ & & & \ddots & h_{r-1,r} & \cdots & h_{r-1,m} \\ & & & & x^{e_r} & \cdots & h_{rm} \end{bmatrix} \in \mathsf{R}^{r\times m},$$

with

- x^{e_i} divides h_{ij}, $1 \leq i < j \leq m$, and
- $0 \leq e_1 \leq e_2 \leq \cdots \leq e_r$.

Notice that a matrix in triangular Smith form can be transformed to Smith form by postmultiplying by a unit upper triangular matrix. We remark that the approach of many Smith form algorithms (see Kaltofen et al., 1990; Giesbrecht, 1995; Villard, 1995) is to randomly "precondition" an input matrix so that it is left equivalent to a triangular Smith form, thus reducing the problem of computing the Smith form to that of computing a Hermite form. In Section 4 we show how to deterministically compute a permutation matrix P such that AP is left equivalent to a triangular Smith form.

On the one hand, because of the uniqueness of the Smith form, a triangular Smith form has the minimum number of nonzero rows of any Hermite form of R. On the other hand, any triangular Smith form is actually in Howell form, which has a maximum number of nonzero rows of any Hermite form of A. Thus, unlike the example in (3), every other Hermite form of a matrix H in triangular Smith form has the same number of nonzero rows as H. Moreover, the two divisibility conditions of Definition 2 can be used to show the following result.

Lemma 1. If $A \in \mathsf{R}^{n\times m}$ is left equivalent to a triangular Smith form H, then every Hermite form of A is equal to H.

For more details on echelon forms over rings, also principal ideal rings with zero divisors, see (Storjohann, 2000, Section 1.4).

3. The x-Hermite decomposition

Definition 3. An *x-Hermite basis* of a full column rank $A \in \mathsf{K}[x]^{n\times m}$ is a matrix

$$H := \begin{bmatrix} x^{e_1} & v_1^{[2]} & v_1^{[3]} & \cdots & v_1^{[m]} \\ & x^{e_2} & v_2^{[3]} & \cdots & v_2^{[m]} \\ & & x^{e_3} & & \vdots \\ & & & \ddots & v_{m-1}^{[m]} \\ & & & & x^{e_m} \end{bmatrix}$$

such that

- $e_i \in \mathbb{Z}_{\geq 0}$ and the offdiagonal entries $v_1^{[i]}, \ldots v_{i-1}^{[i]}$ in column i of H have degree strictly less than e_i, $1 \leq i \leq m$, and
- the matrix $U := AH^{-1}$ is over $\mathsf{K}[x]$ and $\mathrm{Rem}(U, x) \in \mathsf{K}^{n\times m}$ has full column rank over K.

We call $A = UH$ the x-Hermite decomposition of A. Let

$$v^{[i]} := \begin{bmatrix} v_1^{[i]} \cdots v_{i-1}^{[i]} \end{bmatrix}^T \in \mathsf{K}[x]^{(i-1)\times 1}$$

be the column vector of strictly offdiagonal entries in column i of H. Because H is upper triangular, it can be expressed as the product of structured matrices as follows:

$$H = \prod_{i=1}^{m} \left[\begin{array}{c|c|c} I_{m-i} & v^{[m-i+1]} & \\ \hline & x^{e_{m-i+1}} & \\ \hline & & I_{i-1} \end{array}\right]$$

$$= \overset{H_m}{\begin{bmatrix} 1 & & & & v_1^{[m]} \\ & 1 & & & v_2^{[m]} \\ & & 1 & & v_3^{[m]} \\ & & & \ddots & \vdots \\ & & & & x^{e_m} \end{bmatrix}} \cdots \overset{H_3}{\begin{bmatrix} 1 & & v_1^{[3]} & & \\ & 1 & v_2^{[3]} & & \\ & & x^{e_3} & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}} \overset{H_2}{\begin{bmatrix} 1 & v_1^{[2]} & & & \\ & x^{e_2} & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}} \overset{H_1}{\begin{bmatrix} x^{e_1} & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}} \qquad (4)$$

This gives rise to the decomposition

$$H^{-1} = \prod_{i=1}^{m} \left[\begin{array}{c|c|c} I_{i-1} & -v^{[i]}/x^{e_i} & \\ \hline & 1/x^{e_i} & \\ \hline & & I_{m-i} \end{array}\right] = H_1^{-1} H_2^{-1} H_3^{-1} \cdots H_m^{-1}$$

for H^{-1}. The following theorem establishes existence and uniqueness of the x-Hermite decomposition. The algorithm we present for computing the decomposition is based on the proof.

Theorem 4. *Every $A \in \mathsf{K}[x]^{n\times j}$ of full column rank j has a unique x-Hermite decomposition.*

Proof. We use induction on j. The base case $j = 0$ is trivial: $A \in \mathsf{K}[x]^{n\times 0}$ has x-Hermite basis the 0×0 matrix. For $j \geq 1$, our goal is to show that a matrix $\left[\begin{array}{c|c} A & w \end{array}\right] \in \mathsf{K}[x]^{n\times j}$ of rank j, where $A \in \mathsf{K}[x]^{n\times (j-1)}$ and $w \in \mathsf{K}[x]^{n\times 1}$, has a unique x-Hermite basis. Assume, by induction, that $A \in \mathsf{K}[x]^{n\times(j-1)}$ has a unique x-Hermite decomposition $A = UH$. Since $\mathrm{Rem}(U, x) \in \mathsf{K}^{n\times(j-1)}$ has full column rank, U has a submatrix of dimension $j-1$ that is nonsingular modulo x. Assume, up to a row permutation and without loss of generality, that the principal $(j-1)\times(j-1)$ submatrix of U is nonsingular modulo x.

Then we can decompose U and w as

$$\left[\begin{array}{c|c} U & w \end{array}\right] = \left[\begin{array}{c|c} U_1 & w_1 \\ \hline U_2 & w_2 \end{array}\right]$$

with $x \perp \det U_1$. By induction we have $U = AH^{-1}$ over $\mathsf{K}[x]$ with the columns of U linearly independent modulo x. To complete the proof we need to show the existence of unique $e_j \in \mathbb{Z}_{\geq 0}$ and $v^{[j]} \in \mathsf{K}[x]^{(j-1)\times 1}$ of degree bounded by $e_j - 1$ such that the $n \times j$ matrix defined by

$$\begin{aligned} \left[\begin{array}{c|c} A & w \end{array}\right] \left[\begin{array}{c|c} H & v^{[j]} \\ \hline & x^{e_j} \end{array}\right]^{-1} &= \left[\begin{array}{c|c} AH^{-1} & w \end{array}\right] \left[\begin{array}{c|c} I_{j-1} & v^{[j]} \\ \hline & x^{e_j} \end{array}\right]^{-1} \\ &= \left[\begin{array}{c|c} U_1 & (w_1 - U_1 v^{[j]})/x^{e_j} \\ \hline U_2 & (w_2 - U_2 v^{[j]})/x^{e_j} \end{array}\right] \end{aligned} \tag{5}$$

satisfies the following two conditions: (a) the matrix is over $\mathsf{K}[x]$; (b) the matrix taken modulo x has full column rank. Equation (5) and condition (a) are satisfied if and only if $Uv^{[j]} \equiv w \bmod x^{e_j}$. Since $x \perp \det U_1$, conditions (a) and $\deg v^{[j]} < e_j$ are satisfied if and only if $v^{[j]} = \mathrm{Rem}(U_1^{-1}w_1, x^{e_j})$ and $U_2 v^{[j]} \equiv w_2 \bmod x^{e_j}$. Condition (b) is satisfied in addition to (a) if and only if e_j is chosen maximal such that $U_2\mathrm{Rem}(U_1^{-1}w_1, x^{e_j}) \equiv w_2 \bmod x^{e_j}$ and $U_2\mathrm{Rem}(U_1^{-1}w_1, x^{e_j+1}) \not\equiv w_2 \bmod x^{e_j+1}$. To see this last claim, consider taking the matrix in (5) modulo x to obtain a scalar matrix

$$\left[\begin{array}{c|c} \bar{U}_1 & \bar{z}_1 \\ \hline \bar{U}_2 & \bar{z}_2 \end{array}\right].$$

Then $U_2\mathrm{Rem}(U_1^{-1}w_1, x^{e_j+1}) \not\equiv w_2 \bmod x^{e_j+1}$ if and only if the the transformed matrix

$$\left[\begin{array}{c|c} \bar{U}_1 & \bar{z}_1 \\ \hline \bar{U}_2 & \bar{z}_2 \end{array}\right] \left[\begin{array}{c|c} I_{j-1} & -\bar{U}_1^{-1}\bar{z}_1 \\ \hline & 1 \end{array}\right] = \left[\begin{array}{c|c} \bar{U}_1 & \\ \hline \bar{U}_2 & \bar{z}_2 - \bar{U}_2\bar{U}_1^{-1}\bar{z}_1 \end{array}\right]$$

has full column rank, that is, $\bar{z}_2 - \bar{U}_2\bar{U}_1^{-1}\bar{z}_1$ is not the zero vector. Because w is linearly independent on the columns of U, such a maximal e_j does exist. □

Algorithm `XHermiteDecomposition` is shown in Figure 1. We start with the trivial decomposition $A = UH$ where $U = A$ and $H = I_m$. Loop iteration j computes H_j (see (4)) and updates the decomposition using the identity $A = UH = (UH_j^{-1})(H_j H)$. Phases 1 and 2 use the linear x-adic lifting (Dixon, 1982; Moenck and Carter, 1979) (see also Mulders and Storjohann, 2004, Section 5) to obtain e_j and $v^{[j]} = \mathrm{Rem}(U_1^{-1}w_1, x^{e_j})$. To achieve a good cost, phase 1 uses x^d-adic lifting as far as possible. If $e_j = td + s$ for $0 \leq s < d$, then phase 1 uses t steps of x^d-adic lifting to find the maximal t such that $U_2\mathrm{Rem}(U_1^{-1}w_1, x^{td}) \equiv w_2 \bmod x^{td}$. Phase 2 does a single x^s-adic lifting step, and at the same time determines a row permutation matrix Q to ensure that the principal $j \times j$ submatrix of U will be nonsingular modulo x for the next loop iteration. The permutations Q at each phase are recorded in a permutation P, initialized to be I_n,

```
XHermiteDecomposition($A, n, m, d$)
Input: Full column rank $A \in \mathsf{K}[x]^{n\times m}$ with $d = \deg A$.
Output: $U, H$, the $x$-Hermite decomposition $A = UH$.

Initialize $P := I_n$, $U := A$, $H := I_m$ and $B$ to be the $0 \times 0$ matrix.
for $j$ from 1 to $m$ do
   Decompose $PU = \left[\begin{array}{c|c|c} U_1 & w_1 & * \\ \hline U_2 & w_2 & * \end{array}\right]$ where $U_1$ is $(j-1)\times(j-1)$ and $w_1 \in \mathsf{K}[x]^{(j-1)\times 1}$.
   Initialize $v^{[j]}$ to be the $(j-1)\times 1$ zero vector and $u_1, u_2 := w_1, w_2$.
 (1) [Perform linear $x^d$-adic lifting until an inconsistency is found.]
     $v := \mathrm{Rem}(Bu_1, x^d)$;
     for $e_j$ from 0 by $d$ while $x^d \mid (u_2 - U_2 v)$ do
        $u_1, u_2 := (u_1 - U_1 v)/x^d, (u_2 - U_2 v)/x^d$;
        $v^{[j]} := v^{[j]} + v x^{e_j}$;
        $v := \mathrm{Rem}(Bu_1, x^d)$;
     od
 (2) [Perform partial lifting step and determine row swap.]
     $s :=$ the trailing degree of $u_2 - U_2 v$;
     $i :=$ the index of an element of $u_2 - U_2 v$ with trailing degree $s$;
     $Q :=$ the $n \times n$ permutation matrix that swaps row $j$ with row $j + i$;
     $v := \mathrm{Rem}(v, x^s)$;
     $u_1, u_2 := (u_1 - U_1 v)/x^s, (u_2 - U_2 v)/x^s$;
     $v^{[j]}, e_j := v^{[j]} + v x^{e_j}, e_j + s$;
     Comment $v^{[j]}$ and $e_j$ are now as in $H_j$ in (4).
 (3) [Update decomposition using identity $A = UH = (UH_j^{-1})(H_j H)$.]
     Replace column $j$ of $H$ with $\left[\begin{array}{c|c|c} (v^{[j]})^T & x^{e_j} & \end{array}\right]^T$.
     Replace column $j$ of $U$ with $P^{-1}\left[\begin{array}{c|c} u_1^T & u_2^T \end{array}\right]^T$.
     $P := QP$;
 (4) [Update $B$ to be the inverse modulo $x^d$ of the principal $j \times j$ submatrix of $U$.]
     Let the principal $j \times j$ submatrix of $PU$ be $\left[\begin{array}{c|c} * & c \\ \hline r & a \end{array}\right]$ where $a \in \mathsf{K}[x]$.
     $p := \mathrm{Rem}((a - rBc)^{-1}, x^d)$;
     $B := \left[\begin{array}{c|c} \mathrm{Rem}(B + BcprB, x^d) & \mathrm{Rem}(-Bcp, x^d) \\ \hline \mathrm{Rem}(-prB, x^d) & p \end{array}\right] \in \mathsf{K}[x]^{j\times j}$;
od
return $H$, $U$;
```

Fig. 1. Algorithm `XHermiteDecomposition`

so that they can be applied at the start of the next iteration to ensure the principal $(j-1)\times(j-1)$ submatrix of U is nonsingular modulo x. At the start of loop iteration j, matrix B is the inverse modulo x^d of $U_1 \in \mathsf{K}[x]^{(j-1)\times(j-1)}$. Phase 4 updates the inverse using the standard formula.

The following lemma will be useful to bound the cost of the algorithm.

Lemma 2. Let $A = UH$ be the x-Hermite decomposition of $A \in \mathsf{K}[x]^{n\times m}$. If $\deg A \leq d$,then $\deg U \leq d$ and $\deg\det H \leq md$.

Proof. Because of the triangular shape and degree properties of H, the matrix H^{-1} is a proper matrix fraction: for every entry in H^{-1}, the numerator has degree less than or equal to the degree of the denominator. By (Kailath, 1980, Lemma 6.3-10), properness of H^{-1} together with the identity $U = AH^{-1}$ implies $\deg \mathrm{Col}(U, j) \leq \deg \mathrm{Col}(A, j)$ for all j.

Now assume, without loss of generality, that the principal $m \times m$ submatrix U_1 of U is nonsingular modulo x. The principal $m\times m$ submatrix A_1 of A is given by $A_1 = U_1 H$. Since $\deg\det A_1 \leq md$ we must have $\deg\det U_1 + \deg\det H \leq md$. □

Correctness of Algorithm `XHermiteDecomposition` follows from Theorem 4 and the previous discussion. Now consider the running time. Each of the m updates of the modular inverse B in phase 4 costs $O(m^2\,\mathsf{M}(d))$ operations from K. Since $\sum_{i=1}^{m} e_i$ is bounded by md (Lemma 2), the total number of x^d-adic lifting steps over all iterations will be bounded by $O(m)$. Noting that each lifting step costs $O(nm\,\mathsf{M}(d))$ field operations from K, we obtain the following result.

Theorem 5. *Algorithm* `XHermiteDecomposition` *is correct. The cost of the algorithm is $O(nm^2\,\mathsf{M}(d))$ operations from* K.

4. Triangular forms over $\mathsf{R} = \mathsf{K}[x]/(x^d)$

Given a full column rank $A \in \mathsf{K}[x]^{n\times m}$, Algorithm `XHermiteDecomposition` in the previous section computed the x-Hermite basis of A in m iterations, column by column. Our algorithm in the next section incorporates matrix multiplication by using only $O(\log n)$ iterations, each iteration computing a block of columns of the x-Hermite basis of AP, where P, the product of the permutation matrices computed over all iterations, is such that the x-Hermite basis of AP is in triangular x-Smith form. In this section we develop the algorithm used to construct P.

Recall that ϕ_d is the homomorphism which maps from $\mathsf{K}[x]$ to $\mathsf{R} = \mathsf{K}[x]/(x^d)$, defined as $\phi_d(a) = \mathrm{Rem}(a, x^d)$. The following lemma shows that part of the x-Hermite decomposition of a full column rank input matrix A over $\mathsf{K}[x]$ can be recovered by computing a Hermite form of $\phi_d(A)$ over $\mathsf{K}[x]/(x^d)$, provided that $\phi_d(A)$ is left equivalent to a triangular Smith form.

Lemma 3. Let $A \in \mathsf{K}[x]^{n\times m}$ have full column rank with x-Hermite decomposition $A = UH$, and let $\bar{H} \in \mathsf{R}^{n\times m}$ be the Hermite form of $\bar{A} := \phi_d(A) \in \mathsf{R}^{n\times m}$. If $\bar{H}$ is in triangular Smith form over R, then the following diagram commutes:

$$\begin{tikzcd} A \arrow[r, "x\text{-Hermite}"] \arrow[d, "\phi_d"'] & H \arrow[d, "\phi_d"'] \\ \bar{A} \arrow[r, "\text{Hermite}"] & \bar{H} \end{tikzcd}$$

In other words, $\bar{H} = \phi_d(H)$.

Proof. Since $x \perp \det U$, $\phi_d(U)$ is unimodular over R. Moreover, since $A = UH$ over $\mathsf{K}[x]$ we have $\phi_d(A) = \phi_d(U)\phi_d(H)$ over R, and thus $\phi_d(H)$ is left equivalent to $\bar{H}$. By Lemma 1 it will be sufficient to show that $\phi_d(H)$ is in Hermite form over R in order to conclude that $\phi_d(H) = \bar{H}$.

Clearly $\phi_d(H)$ is upper triangular since H is in Hermite form. Let k be the number of nonzero rows of $\bar{H}$. We will show that the $\phi_d(H)$ satisfies the following two conditions.
(a) The last $n - k$ rows of $\phi_d(H)$ are zero.
(b) $\phi_d(H)_{jj} \neq 0$ for $1 \leq j \leq k$.
It will follow from (a) and (b) that $\phi_d(H)$ is in Hermite form over R since the diagonal entries of H, and thus also $\phi_d(H)$, are powers of x, and the normalization conditions $\deg H_{ij} < \deg H_{jj}$, $1 \leq i < j \leq k$, for H over $\mathsf{K}[x]$, together with (b), imply the same conditions hold for $\phi_d(H)$ over R: $\deg \phi_d(H)_{ij} < \deg \phi_d(H)_{jj} < d$ for $1 \leq i < j \leq k$.

First we show that (a) holds. Recall that $\bar{H}$, being in triangular Smith form, is also in Howell form over R and satisfies the following property: if $v \in \mathsf{R}^{1\times m}$ has first k entries zero and is an R-linear combination of rows of $\bar{H}$, then v is an R-linear combination of the subset of rows of $\bar{H}$ that have first k entries zero. But $\bar{H}$ has no nonzero rows that have first k entries zero, so the same must be true for $\phi_d(H)$, and because $\phi_d(H)$ is upper triangular, the last $n - k$ rows of $\phi_d(H)$ must be zero.

Next we show that (b) holds. To arrive at a contradiction, let j be minimal such that $\phi_d(H)_{jj}$ is zero, $1 \leq j \leq k$. Then the submatrix comprised of the first j columns of $\phi_d(H)$ has $j - 1$ nonzero rows and is left equivalent to the submatrix comprised of the first j columns of $\bar{H}$ which is in triangular Smith form with j nonzero rows, a contradiction.

This shows that $\phi_d(H)$ is in Hermite form over R and thus by Lemma 1 is equal to $\bar{H}$. □

Of course, to make use of Lemma 3 to compute a part of the x-Hermite basis of A, we need to ensure that the conditions of the lemma are satisfied. Because greatest common divisors in the ring $\mathsf{R} = \mathsf{K}[x]/(x^d)$ involve only powers of x, corresponding to any matrix A over $\mathsf{K}[x]$ is a permutation P (not necessarily unique) such that $\phi_d(AP)$ is left equivalent to a triangular Smith form.

Definition 6. Let $A \in \mathsf{K}[x]^{n\times m}$. An $m \times m$ permutation matrix P is called a *Smith permutation* for $\phi_d(A) \in \mathsf{R}^{n\times m}$ if the matrix $\phi_d(AP)$ has a Hermite form over R that is in triangular Smith form.

Algorithm `ModSmithPermutation` for computing a permutation matrix P as in Definition 6 is shown in Figure 2. The algorithm recurses on the precision parameter d, which refers to the exponent of x. When $d = 1$, the matrix A only has elements from the field K and we compute an `LSP` decomposition of A using the algorithm of Ibarra et al. (1982). For $d > 1$, the algorithm computes an appropriate permutation P_1 over the

```
ModSmithPermutation($A, n, m, d$)
Input: $A \in \mathsf{K}[x]^{n \times m}$ and $d \in \mathbb{Z}_{\geq 0}$.
Output: $P, r$ such that
  • $P$ is a Smith permutation for $\phi_d(A)$ over $\mathsf{K}[x]/(x^d)$, and
  • $r$ is the number of nonzero invariant factors of $\phi_d(A)$.
Condition: $\deg A < d$.

if $d = 1$ then
    $P :=$ the permutation matrix from the LSP decomposition of $A \in \mathsf{K}^{n \times m}$;
    $r :=$ the number of nonzero rows of $S$;
    return $P^{-1}$, $r$;
else
    $d_1, d_2 := \lfloor d/2 \rfloor, \lceil d/2 \rceil$;
    $P_1, r_1 :=$ ModSmithPermutation($\mathrm{Rem}(A, x^{d_1}), n, m, d_1$);
    $T :=$ an upper triangular matrix over $\mathsf{K}[x]$ with $\phi_d(T) \equiv_L \phi_d(AP_1)$ over $\mathsf{K}[x]/(x^d)$;
    Write $T$ as $\left[\begin{array}{c|c} * & * \\ \hline & C \end{array}\right]$ where $C \in \mathsf{K}[x]^{(n-r_1) \times (m-r_1)}$.
    $P_2, r_2 :=$ ModSmithPermutation($x^{-d_1}C, n - r_1, m - r_1, d_2$);
   return $P_1 \,\mathrm{Diag}(I_{r_1}, P_2)$, $r_1 + r_2$;
fi
```

Fig. 2. Algorithm `ModSmithPermutation`

ring $\mathsf{K}[x]/(x^{\lfloor d/2 \rfloor})$, applies P_1 to the work matrix and partially triangularizes it based on how many Smith invariants of A have degree less than $\lfloor d/2 \rfloor$ (which is given as r_1 by the first recursive call). The remaining part of the work matrix is dealt with by computing another permutation matrix P_2 working over the ring $\mathsf{K}[x]/(x^{\lceil d/2 \rceil})$.

The following technical lemma will be used to bound the cost of the algorithm.

Lemma 4. Let d be power of 2. For some i, $0 \leq i \leq \log_2 d$, let $k_1, \ldots, k_{2^i} \in \mathbb{Z}_{\geq 0}$ be such that $\sum_{j=1}^{2^i} k_j = r$. Then $\mathsf{M}(d/2^i) \sum_{j=1}^{2^i} k_j^{\omega-2} \leq (2^{2-\omega})^i r^{\omega-2}\, \mathsf{M}(d)$.

Proof. By (Storjohann, 2000, Lemma 1.9), we have $a^{\omega-2} + b^{\omega-2} \leq 2^{3-\omega}(a+b)^{\omega-2}$ for any $a, b \in \mathbb{Z}_{\geq 0}$. Using $\mathsf{M}(t/2) \leq (1/2)\mathsf{M}(t)$ now gives that

$$\begin{aligned}
\mathsf{M}(d/2^i) \sum_{j=1}^{2^i} k_j^{\omega-2} &= \mathsf{M}(d/2^i)(k_1^{\omega-2} + \cdots + k_{2^i}^{\omega-2}) \\
&\leq \mathsf{M}(d/2^{i-1})(2^{2-\omega})((k_1 + k_2)^{\omega-2} + \cdots + (k_{2^i-1} + k_{2^i})^{\omega-2}) \\
&\ \ \vdots \\
&\leq \mathsf{M}(d)(2^{2-\omega})^i (k_1 + \cdots + k_{2^i})^{\omega-2}.
\end{aligned}$$

□

Theorem 7. *Algorithm* `ModSmithPermutation` *is correct. The cost of the algorithm is* $O(nmr^{\omega-2}\, \mathsf{M}(d))$ *operations from* K.

Proof. The correctness of the algorithm clearly follows from its design. To simplify the analysis, assume d is a power of 2. Then the execution tree of the algorithm will form a complete binary tree with $(\log d) + 1$ levels. Level zero consists of a root node corresponding to a problem of precision d and output value r, the number of nonzero invariant factors. The two children of the root correspond to problems of precision $d/2$ and output values r_1 and r_2, where $r = r_1 + r_2$. In general, nodes at level i of the tree correspond to problems with precision $d/2^i$, and the number of invariant factors found by solving problems at two siblings will be equal to the number found by their parent. We will bound the number of required operations from K by summing the total cost of all nonrecursive work at the problem corresponding to each node of the execution tree. To simplify the analysis we assume that the matrix dimension of each subproblem at a node of the tree is equal to the upper bound $n \times m$.

First consider a problem corresponding to a node at level $\log_2 d$ (a leaf node of the execution tree) that finds $\bar{r}$ invariant factors. The `LSP` decomposition can be computed in time $O(nm\bar{r}^{\omega-2})$ using the rank sensitive variation of `LSP` decomposition (Ibarra et al., 1982) developed by Jeannerod (2006).

Now consider the problem corresponding the nonbase case at level $i < \log_2 d$ (an internal node of the execution tree) that finds $\bar{r} = \bar{r}_1 + \bar{r}_2$ invariant factors. The cost will be dominated by the computation of T. To compute T, first write the matrix AP_1 using a block decomposition as

$$\left[\begin{array}{c|c} E_1 & * \\ \hline E_2 & * \\ \hline \vdots & \vdots \\ \hline E_{\lceil n/\bar{r}_1 \rceil} & * \end{array}\right] \in \mathsf{R}^{n \times m}, \tag{6}$$

each E_i of dimension $\bar{r}_1 \times \bar{r}_1$, except for possibly $E_{\lceil n/\bar{r}_1 \rceil}$ which may have fewer rows. Using the algorithm supporting (Hafner and McCurley, 1991, Theorem 3.1), compute a unimodular matrix $U \in \mathsf{R}^{2\bar{r}_1 \times 2\bar{r}_1}$ such that

$$U \left[\begin{array}{c} E_1 \\ \hline E_2 \end{array}\right]$$

is upper triangular. The cost of computing U is $O(\bar{r}_1^{\omega} \, \mathsf{M}(d/2^i))$ operations from K. Use U to eliminate block E_2 by premultiplying the matrix in (6) by $\mathrm{Diag}(U, I_{n-2\bar{r}_1})$. The submatrix comprised of rows $\bar{r}_1 + 1, \ldots, 2\bar{r}_1$ of the first $\bar{r}_1$ columns of the work matrix has now been zeroed out. Using $\lceil n/\bar{r}_1 \rceil - 2$ steps, the last $n - 2\bar{r}_1$ rows of the first $\bar{r}_1$ columns of the work matrix can be zeroed in a similar fashion. Using an obvious block decomposition, the total cost of producing T using the method just described is $O(nm\bar{r}_1^{\omega-2} \, \mathsf{M}(d/2^i))$ operations from K. Since $\bar{r}_1 \leq \bar{r}$, the nonrecursive work at an internal node of the execution tree that finds $\bar{r}$ invariant factors is $O(nm\bar{r}^{\omega-2} \, \mathsf{M}(d/2^i))$.

At this point we have shown that there exists an absolute constant c such that the nonrecursive work at a particular node at level i of the execution tree is bounded by

$$cnm\bar{r}^{\omega-2} \, \mathsf{M}(d/2^i), \tag{7}$$

where $\bar{r}$ is the number of invariant factors found. Since the sum of the invariant factors found over all nodes at a particular level is r, we can use Lemma 4 to bound the cost of all nodes at level i by

$$T(i) = (2^{2-\omega})^i cnmr^{\omega-2}\,\mathsf{M}(d). \tag{8}$$

The result now follows by summing the bound (8) over all i:

$$\sum_{i=0}^{\log_2 d} T(i) = cnmr^{\omega-2}\,\mathsf{M}(d) \sum_{i=0}^{\log_d} (2^{2-\omega})^i \leq cnmr^{\omega-2}\,\mathsf{M}(d) \sum_{i=0}^{\infty} (2^{2-\omega})^i \in O(nmr^{\omega-2}\,\mathsf{M}(d)),$$

using the assumption that $\omega > 2$. □

5. Triangular x-Smith decompositions

In this section, we present an algorithm to compute a triangular x-Smith decomposition of a full column rank $A \in \mathsf{K}[x]^{n\times m}$ in $O(nm^{\omega-1}(\log n)^2\,\mathsf{M}(d))$ operations from K. The special structure of the x-Smith form helps us achieve this cost bound as compared to the $O(nm^2\,\mathsf{M}(d))$ cost bound achieved for x-Hermite decomposition in Section 3.

We start with a few definitions and preliminaries.

Definition 8. A triangular x-Smith decomposition of a full column rank matrix $A \in \mathsf{K}[x]^{n\times m}$ is $AP = UH$, where P is an $m \times m$ permutation matrix, $U \in \mathsf{K}[x]^{n\times m}$ is such that $\mathrm{Rem}(U, x)$ has full column rank over K, and H can be written as

$$H = \begin{bmatrix} x^{e_1} & v_1^{[2]} & v_1^{[3]} & \cdots & v_1^{[m]} \\ & x^{e_2} & v_2^{[3]} & \cdots & v_2^{[m]} \\ & & x^{e_3} & & \vdots \\ & & & \ddots & v_{m-1}^{[m]} \\ & & & & x^{e_m} \end{bmatrix} \in \mathsf{K}[x]^{m\times m},$$

with

- $e_i \in \mathbb{Z}_{\geq 0}$ and the offdiagonal entries $v_1^{[i]}, \ldots, v_{i-1}^{[i]}$ in column i of H have degree strictly less than e_i, $1 \leq i \leq m$, and
- $e_1 \leq e_2 \cdots \leq e_m$ and x^{e_i} divides $v_i^{[j]}$, $1 \leq i < j \leq m$.

A matrix H satisfying these properties is said to be in *triangular x-Smith form.*

A triangular x-Smith decomposition always exists and is unique only up to the choice of P. By Lemma 3, the diagonal entries $x^{e_1}, \ldots, x^{e_m}$ of H are the same for any triangular x-Smith decomposition of A and are equal to the diagonal entries of the Smith form of A over $\mathsf{K}[x]/(x^{md+1})$, where $d = \deg A$. We call these diagonal entries the *x-Smith invariants* of A. Note that Lemma 2 gives the degree bounds $\deg U \leq d$ and $\deg\det H \leq md$.

The remaining subsections are organized as follows. In Subsection 5.1 we present an outline of our approach for computing an x-Smith decomposition. The complete algorithm with all computational steps is given and analyzed in Subsection 5.2. In Subsection 5.3 a simple refinement of the algorithm is presented which improves the running time by a factor of $\log\log n$. For simplicity, Subsections 5.1–5.3 assume the input matrix is square and nonsingular. The extension to rectangular inputs of full column rank is straightforward and is given in Subsection 5.4.

5.1. Outline of the algorithm

Our algorithm exploits a degree × dimension compromise. In each iteration we will find a block of columns of a triangular x-Smith form: the dimension of the block decreases by half after each iteration but the working precision doubles.

Let $A \in \mathsf{K}[x]^{n\times n}$ be nonsingular. Without loss of generality, by augmenting A as $\operatorname{diag}(I, A)$ for an identity matrix of dimension at most n, we may assume that $n = \sum_{i=0}^{t} 2^i = 2^{t+1} - 1$ for some $t \in \mathbb{Z}_{\geq 0}$. The next lemma shows how the columns of an x-Smith form can then be partitioned into $t+1$ contiguous blocks from left to right, each having half the previous one: $2^t, 2^{t-1}, \ldots, 1$. This partitioning is illustrated in Figure 3.

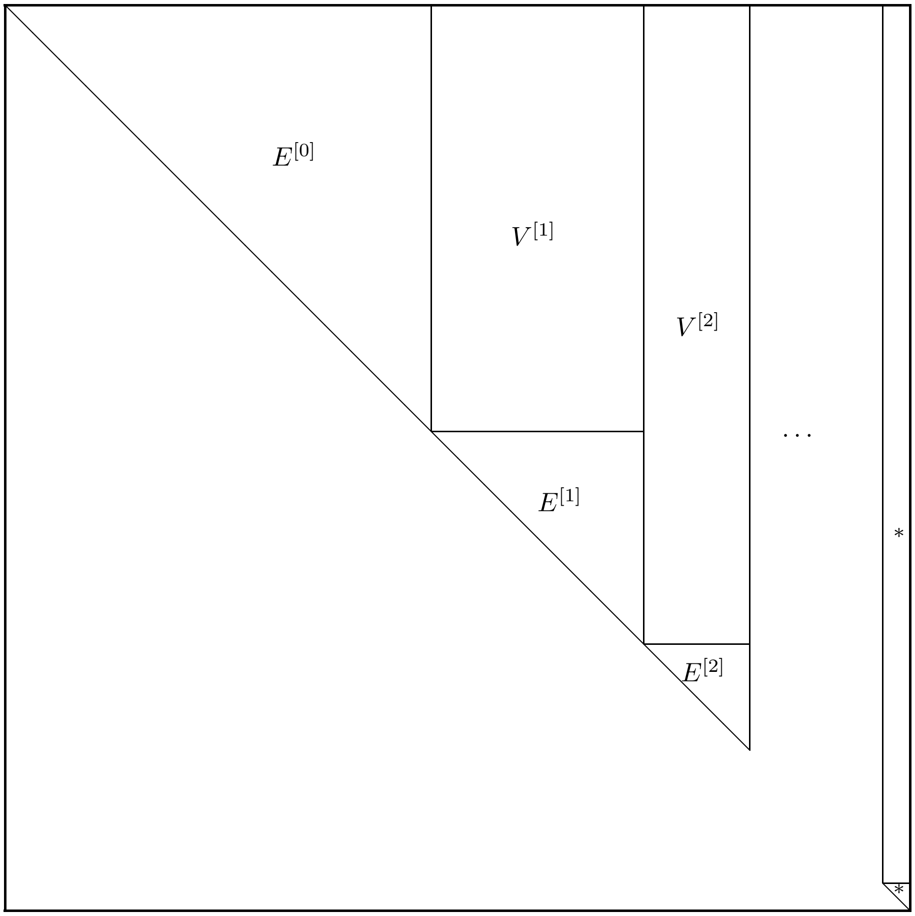

Fig. 3. Partitioning of a triangular x-Smith form

Lemma 5. Assume $n = \sum_{i=0}^{t} 2^i = 2^{t+1} - 1$ for some $t \in \mathbb{Z}_{\geq 0}$, and define $k_i = 2^{t+1} - 2^{t-i+1}$ and $r_i = k_{i+1} - k_i = 2^{t-i}$ for $i = 0, \ldots, t$. Let $H \in \mathsf{K}[x]^{n\times n}$ be a triangular

x-Smith form of a nonsingular $A \in \mathsf{K}[x]^{n\times n}$. Then H can be decomposed as

$$H = \prod_{i=0}^{t} \left[\begin{array}{c|c|c} I_{k_i} & V^{[i]} & \\ \cline{2-3} & E^{[i]} & \\ \hline & & I_{n-k_{i+1}} \end{array}\right], \tag{9}$$

where

$$\left[\begin{array}{c} V^{[i]} \\ \hline E^{[i]} \end{array}\right] = \left[\begin{array}{c|c|c} v_1^{[k_i+1]} & \cdots & v_1^{[k_{i+1}]} \\ \vdots & & \vdots \\ \hline x^{e_{k_i+1}} & & \vdots \\ & \ddots & \\ & & x^{e_{k_{i+1}}} \end{array}\right] \in \mathsf{K}[x]^{k_{i+1}\times 2^{t-i}}.$$

Furthermore, if $\deg A = d$ then $\deg E^{[i]}, \deg V^{[i]} \le 2^{i+1}d$ for all $0 \le i \le t$.

Proof. The decomposition of H in (9) is clearly correct, and we know from Definition 8 that $\deg V^{[i]} \le \deg E^{[i]}$. It thus remains to establish the claimed bound for $\deg E^{[i]}$.

Since $\det H$ is a divisor of $\det A$ we have $\sum_{i=1}^{n} e_i \le \deg\det A \le nd$. Assume, to arrive at a contradiction, that $\deg E^{[i]} \ge 2^{i+1}d + 1$. Then, because each diagonal entry in H divides the next, the last diagonal entry $x^{e_{k_{i+1}}}$ in $E^{[i]}$ must have $e_{k_{i+1}} \ge 2^{i+1}d + 1$. But then $\sum_{j=1}^{n} e_j \ge \sum_{j=k_{i+1}}^{n} e_j \ge \sum_{j=k_{i+1}}^{n}(2^{i+1}d + 1) > nd$, a contradiction. □

Figure 4 presents our approach to compute a triangular x-Smith decomposition based on the decomposition and degree bounds in Lemma 5. Parts A and B in the loop clearly demarcate the operations done over $\mathsf{K}[x]/(x^{2^{i+1}d+1})$ and over $\mathsf{K}[x]$, respectively.

The approach can be understood by considering the first two iterations. Consider the first iteration $i = 0$. Part A computes a Smith permutation Q_0 of A over the ring $\mathsf{K}[x]/(x^{2d+1})$, together with the Hermite form of A over $\mathsf{K}[x]/(x^{2d+1})$. Note that the computation of Q_0 requires considering all columns of A; indeed, by definition, the first Smith invariant of A over R is the gcd of all entries of A. By Lemmas 3 and 5, the precision $2d + 1$ will be sufficient to capture at least the first $k_1 = 2^t$ x-Smith invariants of A over $\mathsf{K}[x]$. Thus, in part B we work over $\mathsf{K}[x]$ and discard all but the first k_1 columns of the Hermite form computed in part A, replacing the last $n - k_1$ columns with the same columns of I_n in order to obtain H_0. To complete iteration $i = 0$ we set $P^{(1)} = Q_0$, $H^{(1)} = H_0$, and $U^{(1)} = AP^{(1)}H_0^{-1}$ to obtain the decomposition

$$AP^{(1)} = U^{(1)} \overset{H^{(1)}}{\left[\begin{array}{c|c} E^{[0]} & \\ \hline & I \end{array}\right]}.$$

Now consider the second iteration $i = 1$. Instead of working with A, we can work with $U^{(1)}$ to recover the next block of x-Smith invariant factors. The precision is increased to $4d + 1$, which by the degree bound in Lemma 5 will be sufficient to capture the first $k_2 = 2^t + 2^{t-1}$ x-Smith invariants of $U^{(1)}$. Because the precision has approximately doubled, we need to reduce the dimension of the problem for the Smith permutation and

```
Initialize $P^{(0)} := I_n$, $U^{(0)} := A$, and $H^{(0)} := I_n$.
for $i$ from 0 to $t$ do
A [Working over $\mathsf{R} = \mathsf{K}[x]/(x^{2^{i+1}d+1})$]
    Let $k_i := 2^{t+1} - 2^{t-i+1}$ and $r_i := 2^{t-i}$.
    Compute $Q_i := \mathrm{Diag}(I_{k_i}, *)$, a Smith Permutation of $U^{(i)}$.
    Compute the Hermite form
        $$\left[\begin{array}{c|c|c} I_{k_i} & V^{[i]} & * \\ \hline & E^{[i]} & * \\ \hline & & * \end{array}\right]$$
    of $U^{(i)}Q_i$.
B [Working over $\mathsf{K}[x]$]
    Let $H_i = \left[\begin{array}{c|c|c} I_{k_i} & V^{[i]} & \\ \hline & E^{[i]} & \\ \hline & & I_{r_i} \end{array}\right] \in \mathsf{K}[x]^{n\times n}$.
    Set $(P^{(i+1)}, H^{(i+1)}, U^{(i+1)}) := (P^{(i)}Q_i, H_iH^{(i)}, U^{(i)}Q_iH_i^{-1})$.
    At this point the following invariants hold:
    · $AP^{(i+1)} = U^{(i+1)}H^{(i+1)}$.
    · The first $k_{i+1} = k_i + 2^{t-i}$ columns of $\mathrm{Rem}(U^{(i+1)}, x)$ have full rank over $\mathsf{K}$.
    · The principal $k_{i+1}$ submatrix of $H^{(i+1)}$ is the $x$-Hermite basis of the first $k_{i+1}$
      columns of $AP^{(i+1)}$.
    · The first $k_{i+1}$ principal diagonal entries of $H^{(i+1)}$ are the first $k_{i+1}$ $x$-Smith
      invariants of $A$.
od
```

Fig. 4. Computing a triangular x-Smith decomposition

Hermite form computation over R. To achieve this reduction in dimension, we can exploit the fact that the first k_1 columns of $\mathrm{Rem}(U^{(1)}, x)$ are known to have full column rank over K; in particular, the first k_1 entries in the Hermite form of $U^{(1)}$ over R are known *a priori* to be trivial. This is illustrated more concretely in phases 1 and 2 of Algorithm `TriangularXSmithDecomposition` detailed in the next subsection.

Now consider part B for iteration $i = 1$. We are starting with the decomposition

$$AP^{(1)} = U^{(1)}H^{(1)} \in \mathsf{K}[x]^{n\times n} \tag{10}$$

and want to obtain the decomposition $AP^{(2)} = U^{(2)}H^{(2)} \in \mathsf{K}[x]^{n\times n}$.

Multiplying both sides of (10) by Q_1 on the right, and inserting $Q_1H_1^{-1}H_1Q_1^{-1} = I$ gives

$$\begin{aligned} A\,\overbrace{P^{(1)}Q_1}^{P^{(2)}} &= U^{(1)}Q_1H_1^{-1}H_1 \overset{Q_1^{-1}}{\left[\begin{array}{c|c} I_{k_1} & \\ \hline & * \end{array}\right]} \overset{H^{(1)}}{\left[\begin{array}{c|c} E^{[0]} & \\ \hline & I_{n-k_1} \end{array}\right]} \overset{Q_1}{\left[\begin{array}{c|c} I_{k_1} & \\ \hline & * \end{array}\right]} \\ &= \overbrace{U^{(1)}Q_1H_1^{-1}}^{U^{(2)}}\,\overbrace{H_1H^{(1)}}^{H^{(2)}}. \end{aligned} \tag{11}$$

Identity (11) follows from the previous equation using $Q_1^{-1}H^{(1)}Q_1 = H^{(1)}$, which holds due to the block diagonal structures of $H^{(1)}$ and Q_1. The remaining iterations are similar. Induction on i can now be used to show that the assertions in part B hold after every iteration. This gives the following result.

Lemma 6. The scheme given in Figure 4 correctly computes a triangular x-Smith decomposition of a nonsingular $A \in \mathsf{K}[x]^{n\times n}$ with $\deg A = d$.

5.2. The complete algorithm

Building up on our approach in Figure 4, we now present our algorithm to compute a triangular x-Smith decomposition. Algorithm `TriangularXSmithDecomposition` in Figure 5 adds computational details to the approach of Figure 4. While phase 1 and phase 2 are used in maintaining the dimension × precision compromise, phase 3 is the concrete realization of part A of Figure 4 and phase 4 the concrete realization of part B of Figure 4.

Note that in all the iterations the following dimension × precision invariant holds:

$$s \times (n-k) = 2^{i+1} \times (2^{t-i+2} - 1) = O(n).$$

Our cost analysis will assume that $\mathsf{M}(t) = O(t^{\omega-1})$. Our main use of this assumption is the following bound:

$$\mathsf{M}(sd) \le \mathsf{M}(s)\mathsf{M}(d) \in O((n/(n-k))^{\omega-1}\,\mathsf{M}(d)). \tag{12}$$

Phase 1 uses the `LSP` decomposition algorithm of Ibarra et al. (1982) to find a row permutation R such that the principal $k \times k$ submatrix of $\mathrm{Rem}(RU, x)$ is nonsingular over K. This costs $O(n^\omega)$ operations from K.

Phase 2 first applies a unimodular transformation over $\mathsf{R} = \mathsf{K}[x]/(x^{sd+1})$:

$$\left[\begin{array}{c|c} I & W_1' \\ \hline & W_2' \end{array}\right] := \left[\begin{array}{c|c} U_1^{-1} & \\ \hline -U_2U_1^{-1} & I \end{array}\right] \left[\begin{array}{c|c} U_1 & W_1 \\ \hline U_2 & W_2 \end{array}\right].$$

By (Storjohann, 2003, Proposition 15), $W_1' = \mathrm{Rem}(U_1^{-1}W_1, x^{sd+1})$ can be computed using high-order lifting in $O(n^\omega(\log n)\,\mathsf{M}(d))$ operations from K. Now consider the computation $W_2' = \mathrm{Rem}(W_2 - U_2W_1', x^{sd+1})$. The dimension of U_2 is $(n-k)\times k$ and W_1' is $k \times (n-k)$, and using an obvious block decomposition, U_2W_1' can be computed using $O(n(n-k)^{\omega-1})$ operations from $\mathsf{K}[x]/(x^{sd+1})$. Using (12) shows U_2W_1' can be computed in $O(n^\omega\,\mathsf{M}(d))$ operations from K. Thus, phase 2 costs $O((\log n)n^\omega\,\mathsf{M}(d))$ operations from K.

Phase 3 computes a triangular Smith form of the matrix U' over R. First we find a permutation matrix Q such that $U'Q$ is Smith conditioned over R. As the principal $k \times k$

submatrix of U' is the identity, we need to find a permutation matrix for only W_2'. This is accomplished using Algorithm ModSmithPermutation described in the previous section.

First we triangularize using the algorithm supporting (Hafner and McCurley, 1991, Theorem 3.1), and then recover the Hermite form by reducing offdiagonal entries using the index k reduction transform from (Storjohann, 2000, Section 3.2). All of these steps cost $O(n(n-k)^{\omega-1})$ operations from R. Again using (12), we obtain the cost bound $O(n^{\omega}\,\mathsf{M}(d))$ for phase 3.

Phase 4 updates the matrices P, U, H and the dimension k and the index i. Note that $\deg U \le d$ since the entries of H_i^{-1} are proper fractions over $\mathsf{K}(x)$. Since $\det H_i \perp (x-1)$ we can calculate the updated U as described in the algorithm by working modulo $(x-1)^{d+1}$. Since we can invert an element over $\mathsf{K}[x]/((x-1)^{d+1})$ in cost $\mathsf{M}(d)$, the cost of phase 4 is $O(nr^{\omega-1}\ \mathsf{M}(d)) = O(n^{\omega}\ \mathsf{M}(d))$.

The correctness component of the following result follows from Lemma 6, and the cost bound from the above discussion.

Theorem 9. *Algorithm* TriangularXSmithDecomposition *is correct. The cost of the algorithm is* $O(n^{\omega}(\log n)^2\,\mathsf{M}(d))$ *operations from* K*. This cost estimate assumes that* $\omega > 2$ *and* $\mathsf{M}(t) \in O(t^{\omega-1})$*.*

Note that the $(\log n)^2$ factor comes from high-order lifting in phase 2 being used at each of the $\log n$ iterations.

5.3. Achieving a slightly better cost

We now present a small change in Algorithm TriangularXSmithDecomposition to achieve a cost of $O(n^{\omega}(\log n)^2/(\log\log n)\,\mathsf{M}(d))$. The main idea is to increase our dimension × precision invariant a little to

$$s \times (n-k) = O(n(\log n)^{1/(\omega-1)}) \tag{13}$$

and decrease the bound on the total number of iterations to $O(\log n/\log\log n)$ while still carrying out every iteration in cost $O((\log n)n^{\omega}\,\mathsf{M}(d))$. We remark that to achieve the acceleration we had initially set the dimension × precision invariant to $O(n\sqrt{\log n})$. We would like to thank one of the referees for suggesting the more natural bound in (13).

To motivate the improved convergence, consider an input matrix $A \in \mathsf{K}[x]^{n\times n}$ with $\deg A = d$ and x-Smith invariants $1, \ldots, 1, x^{nd}$. We can find all but one column of a triangular x-Smith form of A in the first iteration working over the ring $\mathsf{K}[x]/(x^{2d+1})$. After finding $n-1$ columns of a triangular x-Smith form, we can increase the precision to nd and work over the ring $\mathsf{K}[x]/(x^{nd+1})$. Thus, instead of using $\log n$ iterations as in Algorithm TriangularXSmithDecomposition in Figure 5, we choose the precision s and the dimension r in the algorithm dynamically. The Algorithm ModSmithPermutation, used in phase 3 of the algorithm, outputs r, the number of non zero rows in a triangular Smith form of W_2' over $\mathsf{R} = \mathsf{K}[x]/(x^{sd+1})$. Thus at the end of phase 3, we shall decompose the triangular Smith form as

$$\left[\begin{array}{c|c|c} I_k & V & * \\ \cline{2-3} & E & * \\ \cline{1-2} & & 0 \end{array}\right]$$

where E is $r \times r$. All we need to maintain is the dimension × precision invariant (13).

```
TriangularXSmithDecomposition($A, n, d$)
Input: Nonsingular $A \in \mathsf{K}[x]^{n\times n}$ with $d = \deg A$ and $n = 2^{t+1} - 1$.
Output: $P, U, H$ such that $AP = UH$ is a triangular $x$-Smith decomposition of $A$.

Initialize $k := 0$, $i := 0$, $P := I_n$, $U := A$, and $H := I_n$.
while $k < n$ do
  (1) [Find row permutation.]
      $R :=$ the permutation from the LSP decomposition of $\mathrm{Rem}(U, x)^T$;
      Decompose $RU = \left[\begin{array}{c|c} U_1 & W_1 \\ \hline U_2 & W_2 \end{array}\right]$ where $U_1$ is $k \times k$ with $\mathrm{Rem}(U_1, x)$ nonsingular.
  (2) [Perform high-order lifting and compute Schur complement.]
   (a) $s := 2^{i+1}$;
      $W_1' := \mathrm{Rem}(U_1^{-1} W_1, x^{sd+1})$;   comment: use high-order lifting
      $W_2' := \mathrm{Rem}(W_2 - U_2 W_1', x^{sd+1})$;
      $U' := \left[\begin{array}{c|c} I_k & W_1' \\ \hline & W_2' \end{array}\right]$;
  (3) [Compute Smith permutation and triangular Smith form over $\mathsf{K}[x]/(x^{sd+1})$.]
      $Q, r :=$ ModSmithPermutation($W_2^{'}, n-k, n-k, sd+1$);
   (b) $r := 2^{t-i}$;
      $T' :=$ an upper triangular matrix with $\phi_d(T') \equiv_L \phi_d(W_2' Q)$ over $\mathsf{K}[x]/(x^{sd+1})$;
      $T :=$ a matrix such that $\phi_d(T)$ is the Hermite form of $\phi_d(T')$ over $\mathsf{K}[x]/(x^{sd+1})$;
      Decompose $T$ as $\left[\begin{array}{c|c|c} I_k & V & * \\ \hline & E & * \\ \hline & & * \end{array}\right]$ where $E$ is $r \times r$.
  (4) [Update $P$, $U$ and $H$.]
      $P, H := P\,\mathrm{Diag}(I_k, Q), \left[\begin{array}{c|c|c} I_k & V & \\ \hline & E & \\ \hline & & I_{n-k-r} \end{array}\right] H$;

      $U := \mathrm{Rem}\left( UQ \left[\begin{array}{c|c|c} I_k & V & \\ \hline & E & \\ \hline & & I_{n-k-r} \end{array}\right]^{-1}, (x-1)^{d+1} \right)$;
  (c) $i := i + 1$;
      $k := k + r$;
od
return $P$, $U$, $H$;
```

Fig. 5. Algorithm TriangularXSmithDecomposition

Corollary 1. A triangular x-Smith decomposition of a nonsingular $A \in \mathsf{K}[x]^{n\times n}$ with $\deg A = d$ can be computed using $O(n^\omega (\log n)^2/\log\log n\ \mathsf{M}(d))$ operations from K, if the following modifications are carried out in Algorithm `TriangularXSmithDecomposition`:

- Change line (a) to: $s := \lceil 2n(\log n)^{1/(\omega-1)}/(n-k)\rceil$;
- Delete line (b) and line (c).

This cost estimate assumes that $\omega > 2$ and $\mathsf{M}(t) \in O(t^{\omega-1})$.

Proof. We shall first prove that every iteration of the modified algorithm can still be done in time $O(n^\omega(\log n)\,\mathsf{M}(d))$.

In phase 2, the right hand side W_1 of the system to be solved has small degree. As noted in Theorem 1, despite the slightly increased precision, high-order lifting (Storjohann, 2003, Section 8) can be adapted to compute W_1' in cost $O(n^\omega(\log n)\,\mathsf{M}(d))$. The cost of the multiplication of a $(n-k)\times k$ matrix with a $k\times(n-k)$ matrix to get W_2' is $O(n(n-k)^{\omega-1}\mathsf{M}(sd+1)) = O(n(n-k)^{\omega-1}s^{\omega-1}\,\mathsf{M}(d))$. Using the dimension × precision invariant shows that the that the cost of this step is $O(n(n-k)^{\omega-1}(\log n)\,\mathsf{M}(d)) = O(n^\omega(\log n)\ \mathsf{M}(d))$. Thus phase 2 still has overall cost $O(n^\omega(\log n)\,\mathsf{M}(d))$.

Previously, the cost of phase 3 was $O(n^\omega\,\mathsf{M}(d))$. Using the assumption that $\mathsf{M}(t) \in O(t^{\omega-1})$, a $(\log n)^{1/(\omega-1)}$ factor increase in the precision s shall increase the cost of this phase to $O(n^\omega(\log n)\,\mathsf{M}(d))$. Phase 1 and phase 4 are oblivious to the precision s and hence can be computed in cost $O(n^\omega\,\mathsf{M}(d))$.

Using the new dimension × precision invariant and the dynamic change in the precision s and dimension r, let us now show that the total number of iterations needed to find a triangular x-Smith decomposition is bounded by $O(\log n/\log\log n)$. Let s_i be the value of s in the i-th iteration and let k_i be the corresponding value of k. Then $s_0 = \lceil 2(\log n)^{1/(\omega-1)}\rceil$ and $k_0 = 0$. After i iterations we have found k_{i+1} columns of a triangular x-Smith form. We know that all the entries in the remaining x-Hermite form are divisible by $x^{s_i d}$. Using the determinant bound nd for the sum of the degrees of the remaining x-Smith invariants, we get

$$s_i(n-k_{i+1}) \le n. \tag{14}$$

From the dimension × precision formula

$$(n-k_{i+1}) \ge 2n(\log n)^{1/(\omega-1)}/s_{i+1}. \tag{15}$$

Combining (14) and (15) we get

$$s_{i+1}/s_i \ge 2(\log n)^{1/(\omega-1)}. \tag{16}$$

Using the initial condition that $s_0 \ge 2(\log n)^{1/(\omega-1)}$ with inequality (16) gives

$$s_i \ge 2^{i+1}(\log n)^{(i+1)/(\omega-1)}.$$

Thus, the least i such that $s_i \ge n$ is the ceiling of

$$\frac{(\omega-1)\log n}{(\omega-1)\log 2 + \log\log n} - 1.$$

□

The extra logarithmic factors in the cost bound of Corollary 1 can be partitioned as $(\log n/\log\log n) \times (\log n)$. The $\log n/\log\log n$ factor is the bound for the number of iterations of the improved algorithm, while the $\log n$ factor is coming from the quadratic convergence of high-order lifting: this $\log n$ factor seems difficult to improve on.

5.4. Extension to rectangular inputs

Algorithm `TriangularXSmithDecomposition` is easily modified to handle rectangular inputs $A \in \mathsf{K}[x]^{n\times m}$ with full column rank. Indeed, Subroutine `ModSmithPermutation` was presented for rectangular inputs, and the only change in the cost analysis is that the term n^ω is replaced with $nm^{\omega-1}$, and the logarithmic factors will be in terms of m, thus yielding an overall running time of $O(nm^{\omega-1}(\log n)^2/(\log\log n)\,\mathsf{M}(d))$.

6. Partial linearization

Given a nonsingular $A \in \mathsf{K}[x]^{n\times n}$, Algorithm `TriangularXSmithDecomposition` from the previous section computes a decomposition $A = UHP^{-1}$ where P is a permutation matrix and H is in triangular x-Smith form. Given a column vector $b \in \mathsf{K}[x]^{n\times 1}$, our algorithm in the next section computes $A^{-1}b$ as $P(H^{-1}(U^{-1}b)))$, exploiting the fact that we know $x \perp \det U$ and $x-1 \perp \det H$. A problem is that entries in H may have degree as high as $n \deg A$. In this section we describe a general approach for rewriting H as a new matrix $\bar{H}$ that has dimension bounded by $2n-1$ and degree bounded by $\deg A$. To motivate our approach, consider the linearization of a monic polynomial $f = x^d + f_{d-1}x^{d-1} + f_{d-2}x^{d-2} + \cdots + f_0$ based on its companion matrix:

$$\begin{bmatrix} x^d + f_{d-1}x^{d-1} + f_{d-2}x^{d-2} + \cdots + f_0 \end{bmatrix} \longleftrightarrow \begin{bmatrix} f_{d-1}+x & f_{d-2} & \cdots & f_0 \\ -1 & x & & \\ & \ddots & \ddots & \\ & & -1 & x \end{bmatrix}. \tag{17}$$

The determinant of the 1×1 degree d matrix on the left of (17) will be equal to that of the $d\times d$ degree 1 matrix on the right. Also, the inverse of the matrix on the left of (17) will appear as an entry in the inverse of the matrix on the right (the last entry in the first column). The linearization we describe in this section is similar but monicity is not required and the linearization can be partial. For example, a 1×1 degree ed matrix can be partially linearized to an equivalent $e\times e$ matrix of degree d.

We begin by defining some notation. Let $e \in \mathbb{Z}_{\geq 0}$ and $d \in \mathbb{Z}_{\geq 1}$ be given. For a column vector $v \in \mathsf{K}[x]^{n\times 1}$, let $C_{e,d}(v)$ denote the unique $n\times e$ matrix that satisfies

$$\mathrm{Quo}(v, x^d) = C_{e,d}(v)\begin{bmatrix} 1 \\ x^d \\ \vdots \\ x^{(e-1)d} \end{bmatrix},$$

with all but possibly the last column (if $e>0$) of degree less than d. If $e=0$ then $C_{e,d}(v)$ is the $n\times 0$ matrix, while for $e\geq 1$

$$v = \mathrm{Rem}(v, x^d) + \mathrm{Col}(C_{e,d}(v), 1)x^d + \cdots + \mathrm{Col}(C_{e,d}(v), e)x^{ed}$$

is the x^d-adic series expansion of v, except that the coefficient $\mathrm{Col}(C_{e,d}(v), e)$ of x^{ed} may have degree larger than or equal to d.

Example 1. $C_{3,1}(\left[\begin{array}{c} 2+3x+x^2+5x^3+2x^4 \end{array}\right]) = \left[\begin{array}{c|c|c} 3 & 1 & 5+2x \end{array}\right]$.

Now define structured matrices E_d and B_d as follows:

$$E_d := -x^d\,\mathrm{Col}(I,1) = \left[\begin{array}{c} -x^d \\ \\ \\ \\ \\ \end{array}\right] \quad \text{and} \quad B_d := \left[\begin{array}{ccccc} 1 & & & & \\ -x^d & 1 & & & \\ & -x^d & \ddots & & \\ & & \ddots & 1 & \\ & & & -x^d & 1 \end{array}\right].$$

Note that B_d^{-1} will be the unit lower triangular Toeplitz matrix with x^{id} on the ith subdiagonal. The dimensions of E_d and B_d will be induced by the context.

Lemma 7. Let $v \in \mathsf{K}[x]$, $e \in \mathbb{Z}_{\geq 0}$ and $d \in \mathbb{Z}_{\geq 1}$. Let $c = v$ if $e = 0$, and $c = \mathrm{Rem}(v, x^d)$ if $e > 0$. The matrix

$$\left[\begin{array}{c|c} c & C_{e,d}(v) \\ \hline E_d & B_d \end{array}\right] \in \mathsf{K}[x]^{(e+1)\times(e+1)} \tag{18}$$

is right equivalent to

$$\left[\begin{array}{c|cccc} v & \mathrm{Quo}(v,x^d) & \cdots & & \mathrm{Quo}(v,x^{ed}) \\ \hline & 1 & & & \\ & & \ddots & & \\ & & & & 1 \end{array}\right]. \tag{19}$$

Proof. The matrix in (19) can be obtained from the matrix in (18) by postmultiplying by the following unimodular transformation:

$$\left[\begin{array}{c|c} 1 & \\ \hline -B_d^{-1}E_d & B_d^{-1} \end{array}\right].$$

□

Note that if $e = 0$, then B_d is 0×1, E_d is 0×1, and both matrices (18) and (19) are simply v itself.

Part 1 of the theorem below follows from Lemma 7, and part 2 follows easily from part 1. Part 3 follows directly from the definition of $C_{e,d}(v)$.

Theorem 10. *Let* $A = \left[\begin{array}{c|c|c} v_1 & \cdots & v_m \end{array}\right] \in \mathsf{K}[x]^{n\times m}$, $\bar{e} = (e_1, \ldots, e_m) \in \mathbb{Z}_{\geq 0}^m$ *and* $d \in \mathbb{Z}_{\geq 1}$.

Let $c_i = v_i$ if $e_i = 0$, and $c_i = \mathrm{Rem}(v_i, x^d)$ if $e_i > 0$, $1 \le i \le m$. The matrix

$$D_{\bar{e},d}(A) := \left[\begin{array}{c|c|c||c|c|c} c_1 & \cdots & c_n & C_{e_1,d}(v_1) & \cdots & C_{e_m,d}(v_n) \\ \hline\hline E_d & & & B_d & & \\ \hline & \ddots & & & \ddots & \\ \hline & & E_d & & & B_d \end{array}\right] \in \mathsf{K}[x]^{\bar{n}\times\bar{m}},$$

with $\bar{n} = n + e_1 + \cdots + e_m$ and $\bar{m} = m + e_1 + \cdots + e_m$, satisfies the following properties:

(1) $D_{\bar{e},d}(A)$ is right equivalent to

$$\left[\begin{array}{c||ccc|c|ccc} A & \mathrm{Quo}(v_1, x^d) & \cdots & \mathrm{Quo}(v_1, x^{e_1 d}) & \cdots & \mathrm{Quo}(v_m, x^d) & \cdots & \mathrm{Quo}(v_m, x^{e_m d}) \\ \hline\hline & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ \hline & & & & \ddots & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{array}\right]. \quad (20)$$

(2) If $n = m$ then $\det A = \det D_{\bar{e},d}(A)$, and the principal $n \times n$ submatrix of the adjoint of $D_{\bar{e},d}(A)$ is equal to the adjoint of A.

(3) If $\deg v_i \le (e_i + 1)d$ for $1 \le i \le m$, then $\deg D_{\bar{e},d}(A) \le d$.

We remark that if all components of $\bar{e}$ are identical, the matrix in (20) corresponds to the matrix used for the "reduction to lower order" technique described in (Storjohann, 2006, Section 2). In the context of the minimal approximate basis computation discussed there, the fact that (20) has degree as high as A does not affect the cost of the algorithm since the entries can simply be truncated modulo the working precision x^{2d-1}. The key point of Theorem 10 is that (20) is right equivalent to $D_{\bar{e},d}(A)$ which has degree bounded by d (provided that condition 3 of the theorem holds).

The following corollary of Theorem 10 illustrates the usefulness of the partial linearization to the case of linear algebra problems.

Corollary 2. Suppose $\deg A > 0$ and let the average column degree of A be $d := \lceil (\sum_{i=1}^{m} \deg v_i)/m \rceil$. If each $e_i \in \mathbb{Z}_{\ge 0}$ is chosen minimal such that the condition $\deg v_i \le (e_i + 1)d$ from part 3 of Theorem 10 holds, then $D := D_{\bar{e},d}(A)$ enjoys the following properties:

- $\deg D \le d$.
- D has fewer than m extra columns and m extra rows compared to A.
- $\mathrm{rank}(D) = \mathrm{rank}(A) + e_1 + \cdots + e_m$.
- D has the same Smith form as A up to some additional trivial invariant factors.

Furthermore, if $n = m$ then the following hold:

- $\det A = \det D$.
- The adjoint of A is equal to the principal $n \times n$ submatrix of the adjoint of D.

Proof. The only claim that does not follow directly from Theorem 10 is that about the dimension of D. We have $e_i = 0$ if $\deg v_i = 0$ and $e_i < (\deg v_i)/d$ otherwise, the latter case occurring for at least one column because of the assumption that $\deg A > 0$. It follows that $e_1 + \cdots + e_m < \sum_{i=1}^{m} (\deg v_1)/d \leq m$ so that $\bar{m} < 2m$ and $\bar{n} < n + m$. □

Example 2. For brevity, let us indicate a polynomial of degree t with $[t]$, and consider a 5×5 input matrix with the following degree structure, where zero polynomials are indicated with a blank:

$$A = \left[\begin{array}{ccccc} [0] & & & [5] & [18] \\ & [0] & & [5] & [18] \\ & & [0] & [5] & [18] \\ & & & [6] & [18] \\ & & & & [19] \end{array}\right].$$

The construction of Corollary 2 specifies $d = 5$ and $\bar{e} = (0, 0, 0, 1, 3)$, giving

$$D_{\bar{e},d}(A) = \left[\begin{array}{ccccc||c|ccc} [0] & & & [4] & [4] & [0] & [4] & [4] & [3] \\ & [0] & & [4] & [4] & [0] & [4] & [4] & [3] \\ & & [0] & [4] & [4] & [0] & [4] & [4] & [3] \\ & & & [4] & [4] & [1] & [4] & [4] & [3] \\ & & & & [4] & & [4] & [4] & [4] \\ \hline\hline & & & -x^5 & & 1 & & & \\ & & & & -x^5 & & 1 & & \\ & & & & & & -x^5 & 1 & \\ & & & & & & & -x^5 & 1 \end{array}\right].$$

Example 3. The approach of Theorem 2 can also be used to partially linearize the rows of the input matrix. Let $A \in \mathsf{K}[x]^{m \times n}$ have $\deg A > 0$, and consider the matrix $D := D_{\bar{e},d}(A^T)^T$. The degrees of entries in D will then be bounded by the average of the row degrees of A, and D will satisfy all the properties stated in Theorem 10.

The sum E of the column degrees (or row degrees) gives an *a priori* bound for $\deg \det A$. The partial linearization used in Theorem 2 is particularly effective if $\deg \det A$ is close to E, or even equal to E as in the column reduced matrix in Example 2. However, the technique is not useful if A has, simultaneously, some columns and rows of consistently large degree. We now develop an approach to handle such inputs based on a better bound for $\deg \det A$.

By definition, $\det A = \sum_{\sigma \in S_n} \operatorname{sign}(\sigma) \prod_{i=1}^n A_{i,\sigma_i}$ where S_n is the set of all permutations of $(1, 2, \ldots, n)$. This gives the following *a priori* bound for $\deg \det A$. The bound is tight generically.

Fact 11. $\deg \det A \leq \text{GenericDetBnd}(A) := \max_{\sigma \in S_n} \sum_{i=1}^n \deg A_{i,\sigma_i}$.

Up to a row and column permutation, we may assume that $d_i := \deg A_{i,i}$ bounds the degree of all entries in the submatrix $A_{i \ldots n, i \ldots n}$, $1 \leq i \leq n$. Such a row and column permutation can be found by sorting the set of triples $\{(i, j, \deg A_{i,j})\}_{1 \leq i,j \leq n}$ into nonincreasing order according to their third component. Let $E = d_1 + \cdots + d_n$. Then $E \leq \text{GenericDetBnd}(A)$ by definition. Set $d := \lceil E/n \rceil$ and choose $\bar{e} = (e_1, \ldots, e_n)$ with $e_i \in \mathbb{Z}_{\geq 0}$ minimal such that $d_i \leq (e_i + 1)d$. Now consider the matrix $D_{\bar{e},d}(A)$. By construction, row i of $D_{\bar{e},d}(A)$ will have degree bounded by d_i for $1 \leq i \leq n$, and all other rows will have degree bounded by d. Let $\bar{e}'$ denote $\bar{e}$ augmented with $\sum_i^n e_i$ zeroes. Considering the matrix $D_{\bar{e}',d}(D_{\bar{e},d}(A)^T)^T$ gives the following corollary.

Corollary 3. Let nonsingular $A \in \mathsf{K}[x]^{n \times n}$ with $\deg A > 0$ be given. Using the choices for d, $\bar{e}$ and $\bar{e}'$ as specified above, the matrix $D := D_{\bar{e}',d}(D_{\bar{e},d}(A)^T)^T$ will enjoy the following properties:

- $\deg D \leq \lceil \text{GenericDetBnd}(A)/n \rceil$.
- $\text{Dimension}(D) < 3\,\text{Dimension}(A)$.
- $\det D = \det A$.
- The Smith form of D is equal to $\text{Diag}(I, \text{SmithForm}(A))$.
- The principal $n \times n$ submatrix of D^{-1} is equal to A^{-1}.
- If $b \in \mathsf{K}[x]^{1 \times n}$ then bA^{-1} is equal to the principal $1 \times n$ subvector of $\begin{bmatrix} b & 0 & \cdots & 0 \end{bmatrix} D^{-1}$.

Example 4. Consider an input matrix with the following degree structure:

$$A = \begin{bmatrix} [19] & [1] & [5] & [3] & [19] \\ [4] & [6] & [3] & [6] & [0] \\ [0] & [0] & [0] & [0] & [0] \\ [17] & [6] & [0] & [0] & [0] \\ [19] & [0] & [0] & [0] & [0] \end{bmatrix}$$

The recipe supporting Corollary 3 specifies $d = 5$ and $\bar{e} = (3, 1, 0, 0, 0)$. The column linearization produces

$$D_{\bar{e},d}(A) = \left[\begin{array}{ccccc|ccc|c}
[4] & [1] & [5] & [3] & [19] & [4] & [4] & [4] & \\
[4] & [4] & [3] & [6] & [0] & & & & [1] \\
[0] & [0] & [0] & [0] & [0] & & & & \\
[4] & [4] & [0] & [0] & [0] & [4] & [4] & [2] & [1] \\
[4] & [0] & [0] & [0] & [0] & [4] & [4] & [4] & \\
\hline\hline
-x^5 & & & & & 1 & & & \\
 & & & & & -x^5 & 1 & & \\
 & & & & & & -x^5 & 1 & \\
 & -x^5 & & & & & & & 1
\end{array}\right].$$

Since $\sum_{i=1}^{n} e_i = 4$ we have $\bar{e}' = (3, 1, 0, 0, 0, 0, 0, 0, 0)$, and the row linearization of the above matrix produces

$$D_{\bar{e}',d}(D_{\bar{e},d}(A)^T)^T = \left[\begin{array}{ccccccccc||cccc}
[4] & [1] & [4] & [3] & [4] & [4] & [4] & [4] & & -x^5 & & & \\
[4] & [4] & [3] & [4] & [0] & & & & [1] & & & & -x^5 \\
[0] & [0] & [0] & [0] & [0] & & & & & & & & \\
[4] & [4] & [0] & [0] & [0] & [4] & [4] & [2] & [1] & & & & \\
[4] & [0] & [0] & [0] & [0] & [4] & [4] & [4] & & & & & \\
-x^5 & & & & & 1 & & & & & & & \\
 & & & & & -x^5 & 1 & & & & & & \\
 & & & & & & -x^5 & 1 & & & & & \\
 & -x^5 & & & & & & & 1 & & & & \\
\hline\hline
 & & [0] & & [4] & & & & & 1 & -x^5 & & \\
 & & & & [4] & & & & & & 1 & -x^5 & \\
 & & & & [4] & & & & & & & 1 & \\
\hline
 & & & [1] & & & & & & & & & 1
\end{array}\right].$$

7. Deterministic rational system solving

Let $A \in \mathsf{K}[x]^{n\times n}$ be nonsingular with $d = \deg A$. In this section we apply the tools developed in the previous sections to obtain a deterministic algorithm for rational system solving: given $b \in \mathsf{K}[x]^{n\times 1}$ compute $A^{-1}b$.

We will use Algorithm $\texttt{RationalSol}[X]$ from Storjohann (2003). Given an $X \in \mathsf{K}[x]$ that is relatively prime to $\det A$ and satisfies $\deg X \geq d$, the call $\texttt{RationalSol}[X](A, b)$ will produce $(gv, g) \in (\mathsf{K}[x]^{n\times 1}, \mathsf{K}[x])$ with $Av = b$ and g monic of minimal degree such that gv is over $\mathsf{K}[x]$. By (Storjohann, 2003, Corollary 16), if $\deg X \in O(d)$ and $\deg b \in O(nd)$, the cost of algorithm `RationalSol` will be bounded by $O(n^{\omega}(\log n)\,\mathsf{M}(d) + n\,\mathsf{B}(nd))$ operations from K. If a suitable X is not known *a priori* it can be constructed randomly. Instead of using randomization, Algorithm `RationalSystemSolve` shown in Figure 6 proceeds in three phases. First, an x-basis decomposition $A = UH$ is computed. Second, the system $u := U^{-1}b$ is solved using algorithm `RationalSol`$[X]$ with $X = x^d$. Third, H is partially linearized to a new matrix H' that has degree bounded by d and dimension less than $2n$, and the solution $H^{-1}u$ is computed using `RationalSol`$[X]$ with $X = (x-1)^d$, and with input matrix H' instead of H.

```
RationalSystemSolve($A, b, n, d$)
Input: Nonsingular $A \in \mathsf{K}[x]^{n\times n}$ with $d = \deg A$, $b \in \mathsf{K}[x]^{n\times m}$.
Output: $(gA^{-1}b, g) \in (\mathsf{K}[x]^{n\times 1}, \mathsf{K}[x])$ with $g$ of minimal degree.
  (1) [Compute an $x$-basis decomposition $A = UH$.]
      $P, U, H :=$ TriangularXSmithDecomposition($A, n, d$);
      $H := HP^{-1}$;
  (2) [Solve system $U^{-1}b$.]
      Let $X = x^d$.
      $\bar{u}, g_1 :=$ RationalSol$[X]$($U, b$);
      comment: $U\bar{u} = g_1 b$
  (3) [Solve system $H^{-1}\bar{u}$.]
      Let
      • $\bar{e} = (e_1, e_2, \ldots, e_n)$, where $e_i = \max(0, \lceil \deg(\mathrm{Col}(H, i))/d - 1\rceil)$,
      • $H' = D_{\bar{e},d}(H) \in \mathsf{K}[x]^{(n+e)\times(n+e)}$ where $e = e_1 + \cdots + e_n$, and
      • $\bar{u}' = u$ augmented with $e$ trailing zeroes.
      $X := (x-1)^d$;
      $\bar{v}', g_2 :=$ RationalSol$[X]$($H', \bar{u}'$);
      Let $\bar{v} \in \mathsf{K}[x]^{n\times 1}$ be comprised of the first $n$ entries of $\bar{v}'$.
      return $(\bar{v}, g_1 g_2)$;
```

Fig. 6. Algorithm `RationalSystemSolve`

Theorem 12. *Algorithm* `RationalSystemSolve` *is correct. If* $\deg b \in O(nd)$*, the cost of the algorithm is* $O(n^{\omega}(\log n)^2\,\mathsf{M}(d) + n\,\mathsf{B}(nd))$ *operations from* K*. This cost estimate assumes that* $\omega > 2$ *and* $\mathsf{M}(t) \in O(t^{\omega-1})$*.*

Proof. Correctness of the algorithm follows from the previous discussion and Corollary 2. By Theorem 9, the call to Algorithm `TriangularXSmithDecomposition` completes in the allotted time, and since $\deg U \leq d$, the first call to `RationalSol` completes in the allotted time using the assumption that $\deg b \in O(nd)$ (Storjohann, 2003, Corollary 16). By Cramer's rule, $\deg b \in O(nd)$ implies that $\deg u' \in O(nd)$ also, so the cost of the second call also completes in the allotted time. □

Algorithm `RationalSystemSolve` separates the factorization of A in phase 1 from the linear solving in phases 2 and 3. An algorithm which adjusts a given linear system $Av = b$ by factoring out powers of x from the column space of the system while solving is described by Mulders and Storjohann (2000). If $A \in \mathsf{K}[x]^{n\times m}$ has rank r (which need not be known), and $b \in \mathsf{K}[x]^{m\times 1}$ has degree bounded by rd, then the oracle based solver of Mulders and Storjohann (2000) will find a solution v, or determine that the system is inconsistent, in time $O((n+m)r^2\,\mathsf{B}(d))$.

8. Deterministic row reduction

Let $A \in \mathsf{K}[x]^{n\times n}$ be nonsingular. In this section we give a deterministic algorithm to compute a row reduced form of A. We defer until Subsection 8.1 to recall the definition of a row reduced form. For now, we note that a row reduced form of A is a matrix $R \in \mathsf{K}[x]^{n\times n}$ that is left equivalent to A and has row degrees as small as possible. Thus, row reduction is essentially lattice reduction for polynomial matrices.

Example 5. Let us indicate a polynomial of degree t with $[t]$. The following shows the degree structure in a matrix $A \in \mathsf{K}[x]^{4\times 4}$, a row reduced form R of A, and the unimodular matrix U such that $UA = R$:

$$\overset{U}{\begin{bmatrix} [29] & [29] & [30] & [30] \\ [30] & [30] & [31] & [31] \\ [31] & [31] & [32] & [32] \\ [33] & [33] & [34] & [34] \end{bmatrix}} \overset{A}{\begin{bmatrix} [12] & [13] & [13] & [11] \\ [12] & [13] & [13] & [11] \\ [12] & [14] & [12] & [10] \\ [12] & [14] & [12] & [10] \end{bmatrix}} = \overset{R}{\begin{bmatrix} [0] & [0] & [1] & [0] \\ [2] & [1] & [0] & [1] \\ [1] & [2] & [0] & [2] \\ [1] & [1] & [0] & [4] \end{bmatrix}}.$$

Algorithms for computing a row reduced form of A are given by (Mulders and Storjohann, 2003; Giorgi et al., 2003). The algorithm by Mulders and Storjohann (2003) is deterministic but has cost $O(n^3d^2)$. Modifying the approach of Mulders and Storjohann, which is inherently iterative, to incorporate fast matrix and polynomial multiplication does not seem possible. The difficulty is that, although $\deg R \le \deg A$, the unimodular transformation matrix $U \in \mathsf{K}[x]^{n\times n}$ such that $UA = R$ may have $\deg U \in \Omega(n \deg A)$ (see Example 5).

The algorithm by Giorgi et al. (2003) takes a different approach and achieves an expected running time of $O(n^\omega (\log n)\,\mathsf{B}(d))$. The first step is to compute a segment of the inverse A^{-1} modulo a high power of x. This can be accomplished using high-order lifting, but this requires A to be nonsingular modulo x. For the general case, the indeterminate x is first shifted as $x \to x-\alpha$ for a randomly chosen $\alpha \in \mathsf{K}$ to ensure that x does not divide $\det A$ with high probability. The second phase of the algorithm applies a fast minimal approximant basis algorithm to compute R from the high-order segment of A^{-1}. In this section we show how to derandomize the approach of Giorgi et al. (2003) by first computing an x-basis decomposition $A = UH$, then using the technique of Section 6 to partially linearize H allowing for fast computation of a row reduced form R_1 of H via minimal approximant basis computation, and finally computing a row reduced form R_2 of AR_1^{-1} using the approach of Giorgi et al. (2003) to arrive at a row reduced form R_2R_1 of A.

In Subsection 8.1 we define some notation and recall some basic facts about reduced and minimal approximant bases. Subsection 8.2 gives the deterministic algorithm for row reduction.

8.1. Preliminaries: Reduced basis and minimal approximant basis

Following (Beckermann and Labahn, 1994, Definition 3.1), the *defect* $\mathrm{dct}(w, \vec{n})$ of a row vector

$$w = \begin{bmatrix} w_1 & \cdots & w_m \end{bmatrix} \in \mathsf{K}[x]^{1\times m}$$

with respect to a given multi-index $\vec{n} = (n_1, \ldots, n_m) \in \mathbb{Z}^m$ is defined by

$$\mathrm{dct}(w) = \mathrm{dct}(w, \vec{n}) := \min_i \{n_i + 1 - \deg w_i\}, \tag{21}$$

where the zero polynomial has degree $-\infty$. The notion of defect measures the gap between the degrees of elements of w and the multi-index $\vec{n}$. In particular, the constraints

$$\begin{bmatrix} \overset{\le n_1}{\deg} w_1 & \cdots & \overset{\le n_m}{\deg} w_m \end{bmatrix} \in \mathsf{K}[x]^{1\times m} \tag{22}$$

are satisfied if and only if $\mathrm{dct}(w)$ is positive.

Similar to the definition given by Beckermann et al. (2006), we define the leading coefficient vector $\mathrm{lc}(w, \vec{n}) \in \mathsf{K}^{1\times m}$ of a nonzero $w \in \mathsf{K}[x]^{1\times m}$ with respect to $\vec{n}$ to be the constant coefficient of

$$x^{\mathrm{dct}(w)-1}\, w\, \mathrm{Diag}(x^{-n_1}, \ldots, x^{-n_m}) = \begin{bmatrix} x^{\mathrm{dct}(w)-1-n_1} w_1 & \cdots & x^{\mathrm{dct}(w)-1-n_m} w_m \end{bmatrix}, \tag{23}$$

where we consider the entries as Laurent series. The definition of defect implies that the vector in (23) has degree 0. (We remark that we could equivalently define $\mathrm{dct}(w) = \mathrm{dct}(w, \vec{n})$ to be the unique integer such that the vector in (23) has degree 0.) The leading coefficient of the zero vector is defined to be the zero vector. This definition of leading coefficient extends naturally to matrices. Let $B = \begin{bmatrix} b_1^T & \cdots & b_r^T \end{bmatrix}^T \in \mathsf{K}[x]^{r\times m}$ be a nonzero matrix where each row vector b_i is a row vector of dimension m. Then the leading coefficient $\mathrm{lc}(B, \vec{n}) \in \mathsf{K}^{r\times m}$ of B with respect to $\vec{n}$ is the constant coefficient of the degree 0 matrix

$$\mathrm{Diag}(x^{\mathrm{dct}(b_1)-1}, \ldots, x^{\mathrm{dct}(b_r)-1})\, B\, \mathrm{Diag}(x^{-n_1}, \ldots, x^{-n_m}).$$

Example 6. Since

$$\begin{bmatrix} x^2 & \\ & x^1 \end{bmatrix} \overset{B}{\begin{bmatrix} x^3+2x+1 & 2x \\ 2x^4 & x^5+3x^2 \end{bmatrix}} \begin{bmatrix} x^{-5} & \\ & x^{-6} \end{bmatrix} = \overset{L}{\begin{bmatrix} 1 & \\ 2 & 1 \end{bmatrix}} + \begin{bmatrix} 2x^{-2}+x^{-3} & 2x^{-3} \\ & 3x^{-3} \end{bmatrix},$$

and the defects of the rows of B with respect to $(5, 6)$ are $(3, 2)$, we have $\mathrm{lc}(B, (5, 6)) = L$.

Reduced basis

The following definition and lemma give the essential properties of a reduced basis. For more details we refer to (Beckermann and Labahn, 1997; Beckermann et al., 2006). Recall that $\mathcal{L}(B)$ denotes the set of all $\mathsf{K}[x]$-linear combinations of rows of B.

Definition 13. A matrix $B = \begin{bmatrix} b_1^T & \cdots & b_r^T \end{bmatrix}^T \in \mathsf{K}[x]^{r\times m}$ of rank r is a *reduced basis of type* $\vec{n}$ if each $w \in \mathcal{L}(B)$ admits a unique decomposition $w = \sum_{i=1}^{n} c_i b_i$ with $c_i \in \mathsf{K}[x]$, $\deg c_i \le \mathrm{dct}(b_i) - \mathrm{dct}(w)$, $1 \le i \le n$.

We remark that the notion of reducedness is invariant under a constant shift of the multi-index $\vec{n}$: B is a reduced basis of type $\vec{n} = (n_1, \ldots, n_m)$ if and only if B is a reduced basis of type $(n_1 + c, \ldots, n_m + c)$ for any $c \in \mathbb{Z}$.

Lemma 8. A matrix $B \in \mathsf{K}[x]^{r \times m}$ is a reduced basis of type $\vec{n}$ if and only if the following equivalent conditions are satisfied:

(1) $\mathrm{lc}(B, \vec{n})$ has full row rank r.

(2) If the rows $b_1, \ldots, b_r$ of B are permuted so that their defects are nonincreasing, then $(\mathrm{dct}(b_1), \ldots, \mathrm{dct}(b_r))$ is lexicographically maximal among all bases whose rows are similarly permuted.

Thus, up to row permutation, any two reduced bases of type $\vec{n}$ for the same lattice will have the same tuple of defects. The matrix in Example 6 is evidently reduced with respect to $(5, 6)$ because it satisfies property 1 of Lemma 8. The following fact, which follows from (Kailath, 1980, Lemma 6.3-11), will be useful to obtain degree bounds.

Fact 14. *Let $B \in \mathsf{K}[x]^{n \times n}$ be nonsingular. If either B or B^T is a reduced basis of type $\mathbf{0}_n$ then B^{-1} is a proper matrix fraction, that is,* $\deg((\det B)B^{-1}) \leq \deg \det B$.

The next lemma states an elementary but essential property of reduced bases that appears in various guises (see Beckermann and Labahn, 1994, 1997; Giorgi et al., 2003; Beckermann et al., 2006). Let $\mathbf{1}$ denote the tuple $(1, \ldots, 1)$ of appropriate length.

Lemma 9. Suppose $R_1 \in \mathsf{K}[x]^{n \times n}$ is a reduced basis of type $\vec{n}$, and let $\delta = (\delta_1, \ldots, \delta_n)$ be the defects of the rows of R_1 with respect to $\vec{n}$. If $R_2 \in \mathsf{K}[x]^{n \times n}$ is a reduced basis of type $\delta - \mathbf{1}_n$, then $\mathrm{dct}(\mathrm{Row}(R_2, i), \delta - \mathbf{1}_n) = \mathrm{dct}(\mathrm{Row}(R_2 R_1, i), \vec{n})$, $1 \leq i \leq n$, and $R_2 R_1$ is a reduced basis of type $\vec{n}$.

Proof. By definition, $\mathrm{lc}(R_1, \vec{n})$ is given by the constant coefficient of the degree 0 matrix

$$\mathrm{Diag}(x^{\delta_1 - 1}, \ldots, x^{\delta_n - 1})\, R_1 \,\mathrm{Diag}(x^{-n_1}, \ldots, x^{-n_m}). \tag{24}$$

Similarly, if $\mu = (\mu_1, \ldots, \mu_n)$ are the defects of the rows of R_2 with respect to $\delta - \mathbf{1}_n$, then $\mathrm{lc}(R_2, \delta - \mathbf{1}_n)$ is given by the constant coefficient of the degree 0 matrix

$$\mathrm{Diag}(x^{\mu_1 - 1}, \ldots, x^{\mu_n - 1})\, R_2 \,\mathrm{Diag}(x^{-\delta_1 + 1}, \ldots, x^{-\delta_n + 1}). \tag{25}$$

Premultiplying (24) by (25), and noting by part 1 of Lemma 8 that $\mathrm{lc}(R_1, \vec{n})$ and $\mathrm{lc}(R_2, \delta - \mathbf{1}_n)$ are nonsingular, we may conclude that the matrix

$$\mathrm{Diag}(x^{\mu_1 - 1}, \ldots, x^{\mu_n - 1})\, R_2 R_1 \,\mathrm{Diag}(x^{-n_1}, \ldots, x^{-n_m})$$

has degree 0 with constant coefficient matrix nonsingular. By definition, μ are the defects of the rows of $R_2 R_1$ with respect to $\vec{n}$, and by part 1 of Lemma 8, $R_2 R_1$ is a reduced basis of type $\vec{n}$. □

By *positive part* of a reduced basis we mean the submatrix comprised of the rows with positive defect. All $w \in \mathcal{L}(A)$ that satisfy the degree constraint $\vec{n}$ are generated by the positive part of a reduced basis for A: if $\mathrm{dct}(b_i) \leq 0$ and $\mathrm{dct}(w) > 0$, then the c_i of Definition 13 has $\deg c_i \leq \mathrm{dct}(b_i) - \mathrm{dct}(w) < 0$ and thus c_i is the zero polynomial.

Minimal approximant basis

Let $G \in \mathsf{K}[x]^{n\times m}$, $d \in \mathbb{Z}_{\geq 0}$, and $\vec{n} \in \mathbb{Z}^n$.

Definition 15. An order d *minimal approximant* of type $\vec{n}$ for G is a reduced basis M of type $\vec{n}$ for the lattice $\{w \in \mathsf{K}[x]^{1\times n} \mid wG \equiv 0 \bmod x^d\}$.

Note that a minimal approximant M as in Definition 15 will necessarily have dimension $n \times n$, be nonsingular, and satisfy $MG \equiv 0 \bmod x^d$.

The following theorem, a restatement of (Giorgi et al., 2003, Theorem 2.4), is the main computational tool we require for our deterministic row reduction algorithm.

Theorem 16. *There exists an algorithm* `MinBasis` *that takes as input a tuple* $(G, d, \vec{n}) \in (\mathsf{K}[x]^{n\times m}, \mathbb{Z}_{\geq 0}, \mathbb{Z}^n)$ *and returns as output* $(M, \delta) \in (\mathsf{K}[x]^{n\times n}, \mathbb{Z}^n)$*, an order d minimal approximant M of type $\vec{n}$ for G together with a tuple $\delta = (\delta_1, \ldots, \delta_n)$ of the defects of rows of M. If $m \leq n$, the cost of the algorithm is $O(n^\omega\, \mathsf{B}(d))$ operations in* K.

For brevity, we will say that (M, δ) in Theorem 16 solves the minimal approximant problem with input $(G, d, \vec{n})$. By `PosMinBasis`$(G, d, \vec{n})$ we mean the output of `MinBasis`$(G, d, \vec{n})$ restricted to the rows with positive defect. In general, the output of `PosMinBasis` may be the $0 \times n$ matrix. However, in our application of `PosMinBasis` in algorithm `RowReduce` described in the next subsection, the output will have n rows by construction.

8.2. The algorithm for row reduction

Our deterministic algorithm for computing a row reduced form of a nonsingular input matrix $A \in \mathsf{K}[x]^{n\times n}$ with $\deg A = d$ is shown in Figure 7. Phase 1 computes a triangular x-Smith decomposition $AP = UH$. By multiplying both sides of the equation by P^{-1}, and setting $H \leftarrow HP^{-1}$, we obtain a decomposition $A = UH$ that satisfies the following properties: $x \perp \det U$, $\deg U \leq d$, $\det H$ is a power of x. Furthermore, due to the special degree shape of a triangular x-Smith form, the matrix H will be column reduced: H^T is a (row) reduced basis of type $\mathbf{0}_n$ (see Definition 13).

Phase 2 computes a row reduced form of H. Although H may have some columns as large as nd, a row reduced form of H, as well as the transformation matrix to achieve the form, will have degree bounded by d.

Lemma 10. Let $R_1 \in \mathsf{K}[x]^{n\times n}$ be a row reduced form of H with respect to $(d, \ldots, d)$, and let $U_H \in \mathsf{K}[x]^{n\times n}$ be the unimodular matrix such that $U_H H = R_1$. Then the following degree bounds hold:

(1) $\deg \mathrm{Row}(U_H, i) \leq \deg \mathrm{Row}(R_1, i)$, $1 \leq i \leq n$, and
(2) $\deg U_H \leq \deg R_1 \leq d$.

Proof. Because H is column reduced, H^{-1} is a proper matrix fraction (Fact 14). Considering the identity $U_H = R_1 H^{-1}$ shows the first bound. Now let $V := UU_H^{-1} \in \mathsf{K}[x]^{n\times n}$. Then $A = UH = (UU_H^{-1})(U_H H) = VR_1$. Because R_1 is row reduced, part 2 of Definition 13 gives the bound $\deg R_1 \leq \deg A = d$. □

```
RowReduce($A, n, d$)
Input: Nonsingular $A \in \mathsf{K}[x]^{n\times n}$ with $d = \deg A$.
Output: $R$, a row reduced form of $A$.
  (1) [Compute an $x$-basis decomposition $A = UH$.]
      $P, U, H :=$ TriangularXSmithDecomposition($A, n, d$);
      $H := HP^{-1}$;
  (2) [Compute a row reduced form $R_1$ of $H$.]
      Let
      • $e_i \in \mathbb{Z}_{\geq 0}$ be minimal such that $\deg \mathrm{Col}(H, i) \leq (e_i + 1)d$, $1 \leq i \leq n$,
      • $\bar{e} = (e_1, e_2, \ldots, e_n)$ and $e = e_1 + e_2 + \cdots + e_n$,
      • $D = D_{\bar{e},d}(H) \in \mathsf{K}[x]^{(n+e)\times(n+e)}$,
      • $G = \left[\begin{array}{c|c} \multicolumn{2}{c}{D} \\ \hline -I_n & 0_{n\times e} \end{array}\right] \in \mathsf{K}[x]^{(2n+e)\times(n+e)}$, and
      • $\vec{n} = (\overbrace{d+1, d+1, \ldots, d+1}^{n}, \overbrace{d, d, \ldots, d}^{e}, \overbrace{d, d, \ldots, d}^{n})$.
      $\left[\begin{array}{c|c|c} U_H & S & R_1 \end{array}\right] :=$ PosMinBasis($G, 2d+2, \vec{n}$);
  (3) [Compute a row reduced form $R_2$ of $AR_1^{-1}$.]
      $V := \mathrm{Rem}(AR_1^{-1}, (x-1)^{d+1})$;
      $E := \mathrm{Rem}(\mathrm{Quo}(V^{-1}, x^{(n-1)d+1}, x^{2d+1})$;
      Let
      • $G = \left[\begin{array}{c} E \\ \hline -I_n \end{array}\right]$
      • $\delta = (\delta_1, \delta_2, \ldots, \delta_n, d, \ldots, d)$ with $\delta_i = \mathrm{dct}(\mathrm{Row}(R_1, i), \vec{n})$, $1 \leq i \leq n$.
      $\left[\begin{array}{c|c} R_2 & * \end{array}\right] :=$ PosMinBasis($G, 2d+1, \delta - \mathbf{1}_n$);
      return $R_2 R_1$;
```

Fig. 7. Algorithm `RowReduce`

We refer to (Beckermann et al., 1999, Section 5) and (Beckermann et al., 2006, Section 4) for details on how matrices U_H and R_1 as in Lemma 10 can be recovered by computing a row reduced basis of carefully chosen type $\vec{n}$ for the left kernel of the matrix

$$\left[\begin{array}{c} H \\ \hline -I_n \end{array}\right].$$

The choice of $\vec{n}$ is dictated by *a priori* degree bounds for the rows of U_H and R_1, and the reduced kernel basis itself can be recovered as the positive part of a minimal approximant basis of high enough order. Applying this approach directly is too expensive because $\deg H$ may be large, requiring a minimal approximant basis computation of too high order. Instead, phase 2 of Algorithm `RowReduce` applies the partial linearization technique of Theorem 10 to obtain a minimal approximant problem of order only $2d + 2$.

The statement of the next lemma contains a matrix Q which we first need to define. By Theorem 10, matrix D in phase 2 is right equivalent to the matrix shown in (20) with $A = H$. Applying the same unimodular column transform to the matrix G from phase 2 produces the following matrix

$$\left[\begin{array}{c|c} H & Q \\ \hline & I \\ \hline -I & \end{array}\right] \tag{26}$$

that is right equivalent to G. The matrix Q thus corresponds to the submatrix of the matrix in (20) comprised of the first n rows and last e columns.

Lemma 11. Let G and $\vec{n}$ be as in phase 2 of Algorithm `RowReduce`, and let $Q \in \mathsf{K}[x]^{n\times e}$ be as described above. For any vector $\left[\begin{array}{c|c|c} u_H & s & r_1 \end{array}\right] \in \mathsf{K}[x]^{1\times(n+e+n)}$ of positive defect with respect to $\vec{n}$, we have

$$\left[\begin{array}{c|c|c} u_H & s & r_1 \end{array}\right] \in \mathcal{L}(\texttt{MinBasis}(G, 2d+2, \vec{n}))$$

if and only if $s = -u_H Q$ and $\left[\begin{array}{c|c} u_H & r_1 \end{array}\right] \in \mathcal{L}(\left[\begin{array}{c|c} I & H \end{array}\right])$ with $\mathrm{dct}(r_1, (d, \ldots, d)) > 0$.

Proof. **Only If:** Let $\left[\begin{array}{c|c|c} u_H & s & r_1 \end{array}\right] \in \mathcal{L}(\texttt{MinBasis}(G, 2d+2, \vec{n}))$ have positive defect. Then necessarily $\mathrm{dct}(r_1, (d, \ldots, d)) > 0$. By definition,

$$\left[\begin{array}{c|c|c} u_H & s & r_1 \end{array}\right] G \equiv 0 \bmod x^{2d+2},$$

but since $\deg G \leq d$ and $\deg \left[\begin{array}{c|c|c} u_H & s & r_1 \end{array}\right] \leq d+1$, we can conclude that

$$\left[\begin{array}{c|c|c} u_H & s & r_1 \end{array}\right] G = 0. \tag{27}$$

As noted above, the matrix (26) is right equivalent to G. We conclude that

$$\left[\begin{array}{c|c|c} u_H & s & r_1 \end{array}\right] \left[\begin{array}{c|c} H & Q \\ \hline & I \\ \hline -I & \end{array}\right] = 0. \tag{28}$$

Clearly, (28) implies that $s = -u_H Q$, $r_1 = u_H H$, and thus $\left[\begin{array}{c|c} u_H & r_1 \end{array}\right] \in \mathcal{L}(\left[\begin{array}{c|c} I & H \end{array}\right])$. The claim about $\mathrm{dct}(r_1)$ follows from the definition of $\vec{n}$. The result follows.

If: Let $s = -u_H Q$ and $\left[\begin{array}{c|c} u_H & r_1 \end{array}\right] \in \mathcal{L}(\left[\begin{array}{c|c} I & H \end{array}\right])$ with $\mathrm{dct}(r_1, (d, \ldots, d)) > 0$. Then (28) evidently holds, which shows that the right equivalent system (27) holds also. An argument similar to that used in the proof of Lemma 10 will show that $\deg u_H \leq \deg r_1$ and hence $\mathrm{dct}(u_H, (d+1, \ldots, d+1)) > 0$. It remains to show that $\mathrm{dct}(s, (d, \ldots, d))$ is positive. Note that each column of Q is equal to $\mathrm{Quo}(\mathrm{Col}(H, j), x^{dt})$ for some $1 \leq j \leq n$ and $t \geq 1$. Thus, each component of s is given by

$$u_H \overbrace{(\mathrm{Col}(H, j) - \mathrm{Rem}(\mathrm{Col}(H, j), x^{dt}))/x^{dt}}^{= \mathrm{Quo}(\mathrm{Col}(H, j), x^{dt})}$$

for some j and t. Note that $u_H \,\mathrm{Col}(H, j)$ is a component of r_1, and hence has degree at most $\deg r_1$, while $\mathrm{Rem}(\mathrm{Col}(H, j), x^{dt})$ has degree at most $td - 1$. Since, as shown above,

$\deg u_H \leq \deg r_1$, this gives

$$\deg u_H(\mathrm{Col}(H,j) - \mathrm{Rem}(\mathrm{Col}(H,j), x^{dt}))/x^{td} \leq (\deg r_1 + td - 1) - td < \deg r_1.$$

The result follows. □

Corollary 4 follows directly from the degree relationship established for u_H and s in the second part of the proof of Lemma 11, and also from the fact that $\mathrm{dct}(r_1, (d, \ldots, d)) = d + 1 - \deg r_1$ (and similarly for u_H and s).

Corollary 4. If $\left[\, u_H \middle| s \middle| r_1 \,\right] \in \mathcal{L}(\texttt{MinBasis}(G, 2d + d, \vec{n}))$ with positive defect, then $\mathrm{dct}(r_1, (d, \ldots, d))$ is strictly less than both $\mathrm{dct}(u_H, (d+1, \ldots, d+1))$ and $\mathrm{dct}(s, (d+1, \ldots, d+1))$.

Theorem 17. *Phase 2 of Algorithm* `RowReduce` *correctly computes U_H and R_1 such that $U_H H = R_1$ with R_1 a reduced basis of H of type $(d, \ldots, d)$.*

Proof. The "only if" direction of Lemma 11 implies that every row in $\texttt{MinBasis}(G, 2d+2, \vec{n})$ lives in $\mathcal{L}(\left[\, I \middle| -Q \middle| H \,\right]$. This shows that $\texttt{MinBasis}(G, 2d + 2, \vec{n})$ can have at most n rows. From the "if" direction of Lemma 11, together with Lemma 10, it follows that $\texttt{MinBasis}(G, 2d + 2, \vec{n})$ has exactly n rows. By Corollary 4, the defect of any row $\left[\, u_H \middle| s \middle| r_1 \,\right]$ of $\left[\, U_H \middle| S \middle| R_1 \,\right]$ will be determined by $\mathrm{dct}(r_1, (d, \ldots, d))$. This shows that $\mathrm{lc}(R_1, (d, \ldots, d))$ is nonsingular, and thus R_1 is a reduced basis according to Lemma 8. Moreover, since this is a minimal approximant basis, up to permuting the rows the defects of rows of R_1 will be lexicographically maximal. We conclude that R_1 must be a reduced basis for H. □

Phase 3 follows almost exactly the approach of (Giorgi et al., 2003, Section 3.3) with the following modifications. First, we avoid randomly shifting $x \to x - \alpha$ for a random $\alpha \in \mathsf{K}$ because we know by construction that x will not divide $\det V$. Second, instead of computing the minimal approximant basis with respect to $(d, \ldots, d)$, we use the multi-index δ indicated by Lemma 9. At the start of phase 3 we have $A = VR_1$ where R_1 is row reduced with respect to $(d, \ldots, d)$. Because R_1 is left equivalent to H, whose determinant is a power of x by definition of a triangular x-Smith from, R_1 can be inverted modulo any power of $x - 1$. Since R_2 is a reduced basis for V, the matrix $R_2 V^{-1}$ is unimodular. Thus, $(R_2 V^{-1}) V R_1 = R_2 R_1$ is left equivalent to A, and Lemma 9 ensures that $R_2 R_1$ will be a reduced basis.

Theorem 18. *Algorithm* `RowReduce` *is correct and has cost $O(n^\omega (\log n)^2 \,\mathsf{M}(d) + n^\omega \,\mathsf{B}(d))$ field operations from K. This cost estimate assumes that $\omega > 2$ and $\mathsf{M}(t) \in O(t^{\omega - 1})$.*

Proof. Correctness of the algorithm follows from the previous discussion. By Theorem 9, phase 1 runs in the allotted time. By Theorem 2, the matrix G in phase 2 will have row dimension strictly less than $3n$, showing phase 2 runs in the allotted time using the algorithm supporting Theorem 16. In phase 3, the matrix V can be computed as $\mathrm{Rem}(A\, \mathrm{Rem}(R_1^{-1}, (x-1)^{d+1}), (x-1)^{d+1})$, where $\mathrm{Rem}(R_1^{-1}, (x-1)^{d+1})$ is found in time $O(n^\omega \,\mathsf{M}(d))$ by first computing $\mathrm{Rem}(R_1, x-1)^{-1} \in \mathsf{K}^{n \times n}$ and then using Newton iteration. The high-order component E of V^{-1} in phase 3 can be computed using the algorithm for integrality certification described in (Storjohann, 2003, Section 11). □

We end this section with a worked example of Algorithm `RowReduce`.

Example 7. Consider the following 3×3 matrix of degree 4 over $\mathbb{Z}[x]/(7)$:

$$A = \left[\begin{array}{ccc} x^4+5x^3+5x^2+2x+3 & 2x^4+2x^3+3x^2+5 & 2x^3+x+1 \\ x^4+4\,x^3+4x^2+5x+2 & 2x^4+3x^3+4x^2+6 & 2x^3+3x^2+2 \\ x^4+x^3+2x^2+4x+2 & 6x^4+6x^3+2x^2+3x+3 & 2x^4+5x^3+x^2+2x+4 \end{array} \right].$$

The defects of the rows of A with respect to the multi-index $(4,4,4)$ are $(1,1,1)$. The x-Hermite decomposition $A = UH$ of A has

$$U = \left[\begin{array}{ccc} x^4+5x^3+5x^2+2x+3 & 2x^4+2x^3+3x^2+5 & 2x^3+x+1 \\ x^4+4x^3+4x^2+5x+2 & 2x^4+3x^3+4x^2+6 & 2x^3+3x^2+2 \\ x^4+x^3+2x^2+4x+2 & 6x^4+6x^3+2x^2+3x+3 & 2x^4+5x^3+x^2+2x+4 \end{array} \right]$$

and

$$H = \left[\begin{array}{ccc} 1 & 0 & 4x^9+6x^8+6x^7+x^6+4x^5+5x^4+x^3+2x^2+3x+3 \\ & 1 & 5x^4+4x^3+2x^2+4 \\ & & x^{10} \end{array} \right].$$

For this example the x-Hermite form H has a generic degree structure and is also in triangular x-Smith form. In phase 2 we linearize H with respect to the target degree 4, resulting in the matrix

$$G = \left[\begin{array}{ccc||cc} 1 & 0 & x^3+2x^2+3x+3 & 6x^3+x^2+4x+5 & 4x+6 \\ 0 & 1 & 4x^3+2x^2+4 & 5 & 0 \\ 0 & 0 & 0 & 0 & x^2 \\ \hline\hline & & -x^4 & 1 & \\ & & & -x^4 & 1 \\ \hline 1 & & & & \\ & 1 & & & \\ & & 1 & & \end{array} \right].$$

The minimal approximant basis computation yields a reduced basis with the following degree structure:

$$\begin{aligned} & \texttt{PosMinBasis}(G, 2\cdot 4+2, (4+1, 4+1, 4+1, 4, 4, 4, 4, 4)) \\ & \quad = \left[\; U_H \middle| S \middle| R_1 \; \right] \\ & \quad = \left[\begin{array}{ccc|cc|ccc} [3] & [3] & [2] & [2] & [2] & [3] & [3] & [3] \\ [2] & [2] & [1] & [1] & [1] & [2] & [2] & [3] \\ [2] & [4] & [1] & [3] & [1] & [2] & [4] & [4] \end{array} \right]. \end{aligned}$$

As per Lemma 10, the degrees of rows in U_H are at most the degree of the corresponding rows in R_1, while degrees of rows in S are strictly less. The reduced basis R_1 of H is

$$R_1 = \begin{bmatrix} 2x^3+4x^2+5x & 5x^3+3x^2+3x+1 & 2x^3+6x^2+6x+4 \\ 6x^2+5x+1 & x^2+2x+2 & 6x^3+x^2+5x+4 \\ 4x^2+3x & 4x^4+6x^2+4x & 6x^4+5x^3+3x^2+4x \end{bmatrix}.$$

The tuple of defects of the rows of R_1 with respect to the multi-index $(4,4,4)$ is $\vec{n} = (2,2,1)$. Phase 3 begins by computing the matrix V such that $A = VR_1$:

$$V = \begin{bmatrix} 4x+6 & 2x+3 & 6 \\ 4x+2 & 2x+2 & 6 \\ 4x+6 & 6x+2 & 0 \end{bmatrix}.$$

The computed reduced basis of V of type $(2,2,1) - \mathbf{1}$ is

$$R_2 = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 5x+4 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

with

$$\mathrm{lc}(R_2,(2,2,1)) = \begin{bmatrix} 1 & 2 & \\ & 5 & \\ & & 1 \end{bmatrix}.$$

The final reduced basis for A is given by

$$R_2R_1 = \begin{bmatrix} 2x^3+2x^2+x+2 & 5x^3+5x^2+5 & x^2+2x+5 \\ 2x^3+4x+4 & 5x^3+4x+1 & 2x^4+x^3+x^2+5x+2 \\ 4x^2+3x & 4x^4+6x^2+4x & 6x^4+5x^3+3x^2+4x \end{bmatrix}.$$

The defects of the rows of R_1 with respect to $(4,4,4)$ are $(2,1,1)$.

9. Conclusions

This paper gives derandomizations of the known Las Vegas reductions to polynomial matrix multiplication for two problems: solving a rational linear system and obtaining a row reduced form of a matrix. Let $A \in \mathsf{K}[x]^{n\times n}$ be a nonsingular polynomial matrix with degrees of entries bounded by d, K an abstract field. We have established that the following two problems can be solved using $(n^\omega d) \times O\tilde{}((\log n + \log d)^2)$ field operations from K.

- Nonsingular Rational System Solving: Given a $b \in \mathsf{K}[x]^{n\times 1}$ with $\deg b \in O(nd)$, compute the rational vector $A^{-1}b \in \mathsf{K}(x)$.
- Row Reduction: Compute a matrix $R \in \mathsf{K}[x]^{n\times n}$ that is row reduced and left equivalent to A.

A canonical form for row reduction is provided by the Popov form (see Kailath, 1980). An algorithm supporting the running time stated above for transforming R to Popov form P, as well as computing the unimodular matrix U such that $A = UP$, has recently been given by Sarkar and Storjohann (2011).

The partial linearization technique of Section 6 is applicable to the case of integer matrices and should be useful for integer matrix computations. Some of the other ideas in this paper also carry over to the case of integer matrices. For example, any nonsingular $A \in \mathbb{Z}^{n\times n}$ can be decomposed as $A = UH$ where 2 does not divide $\det U$ and H is in Hermite form with powers of 2 on the diagonal. The main difficulty to compute such a decomposition deterministically in about the same time as required to multiply together two integer matrices with similar size entries as A is the presence of carries in integer arithmetic. The extension of high-order lifting to integer matrices (Storjohann, 2005) uses a shifted number system which requires the choice of a random shift.

References

Beckermann, B., Labahn, G., 1994. A uniform approach for the fast computation of matrix–type Padé approximants. SIAM Journal on Matrix Analysis and Applications 15 (3), 804–823.

Beckermann, B., Labahn, G., 1997. Recursiveness in matrix rational interpolation problems. Journal of Computational and Applied Math 77, 5–34.

Beckermann, B., Labahn, G., Villard, G., 1999. Shifted normal forms of polynomial matrices. In: Dooley, S. (Ed.), Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'99. ACM Press, New York, pp. 189—196.

Beckermann, B., Labahn, G., Villard, G., 2006. Normal forms for general polynomial matrices. Journal of Symbolic Computation 41 (6), 708–737.

Dixon, J. D., 1982. Exact solution of linear equations using p-adic expansions. Numer. Math. 40, 137–141.

von zur Gathen, J., Gerhard, J., 2003. Modern Computer Algebra, 2nd Edition. Cambridge University Press.

Giesbrecht, M., 1995. Fast computation of the Smith normal form of an integer matrix. In: Levelt, A. (Ed.), Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'95. ACM Press, New York, pp. 110–118.

Giorgi, P., Jeannerod, C.-P., Villard, G., 2003. On the complexity of polynomial matrix computations. In: Sendra, R. (Ed.), Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'03. ACM Press, New York, pp. 135–142.

Hafner, J. L., McCurley, K. S., Dec. 1991. Asymptotically fast triangularization of matrices over rings. SIAM Journal of Computing 20 (6), 1068–1083.

Howell, J. A., 1986. Spans in the module $(\mathbb{Z}_m)^s$. Linear and Multilinear Algebra 19, 67—77.

Ibarra, O., Moran, S., Hui, R., 1982. A generalization of the fast LUP matrix decomposition algorithm and applications. Journal of Algorithms 3, 45–56.

Jeannerod, C.-P., September 2006. LSP Matrix Decomposition Revisited. Tech. Rep. Research Report 2006-28, École normale supérieure de Lyon, LIP.

Jeannerod, C.-P., Villard, G., 2005. Asymptotically fast polynomial matrix algorithms for multivariable systems. Int. J. Control 72 (11), 1359–1367.

Kailath, T., 1980. Linear Systems. Prentice Hall, Englewood Cliffs, N.J.

Kaltofen, E., Krishnamoorthy, M. S., Saunders, B. D., 1990. Parallel algorithms for matrix normal forms. Linear Algebra and its Applications 136, 189–208.

Kaplansky, I., 1949. Elementary divisors and modules. Trans. of the Amer. Math. Soc. 66, 464–491.

Moenck, R. T., Carter, J. H., 1979. Approximate algorithms to derive exact solutions to systems of linear equations. In: Proc. EUROSAM '79, volume 72 of *Lecture Notes in Compute Science*. Springer-Verlag, Berlin-Heidelberg-New York, pp. 65–72.

Mulders, T., Storjohann, A., 2000. Rational solutions of singular linear systems. In: Traverso, C. (Ed.), Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'00. ACM Press, New York, pp. 242–249.

Mulders, T., Storjohann, A., 2003. On lattice reduction for polynomial matrices. Journal of Symbolic Computation 35 (4), 377–401.

Mulders, T., Storjohann, A., 2004. Certified dense linear system solving. Journal of Symbolic Computation 37 (4), 485–510.

Sarkar, S., Storjohann, A., 2011. Normalization of row reduced matrices. In: Leykin, A. (Ed.), Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'11. ACM Press, New York, pp. 297–303.

Storjohann, A., 2000. Algorithms for matrix canonical forms. Ph.D. thesis, Swiss Federal Institute of Technology, ETH–Zurich.

Storjohann, A., 2002. High–order lifting. Extended Abstract. In: Mora, T. (Ed.), Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'02. ACM Press, New York, pp. 246–254.

Storjohann, A., 2003. High–order lifting and integrality certification. Journal of Symbolic Computation 36 (3–4), 613–648, extended abstract in Storjohann (2002).

Storjohann, A., 2005. The shifted number system for fast linear algebra on integer matrices. Journal of Complexity 21 (4), 609–650, festschrift for the 70th Birthday of Arnold Schönhage.

Storjohann, A., 2006. Notes on computing minimal approximant bases. In: Decker, W., Dewar, M., Kaltofen, E., Watt, S. (Eds.), Challenges in Symbolic Computation Software. No. 06271 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.
URL `http://drops.dagstuhl.de/opus/volltexte/2006/776`

Storjohann, A., Mulders, T., 1998. Fast algorithms for linear algebra modulo N. In: Bilardi, G., Italiano, G. F., Pietracaprina, A., Pucci, G. (Eds.), Algorithms — ESA '98. LNCS 1461. Springer Verlag, pp. 139–150.

Villard, G., 1995. Generalized subresultants for computing the Smith normal form of polynomial matrices. Journal of Symbolic Computation 20 (3), 269—286.

Wilkening, J., Yu, J., 2011. A local construction of the Smith normal form of a polynomial matrix. Journal of Symbolic Computation 46 (1), 1–22.