Computing Hermite Forms of Polynomial Matrices

Somit Gupta
somit.gupta@gmail.com

Arne Storjohann
astorjoh@uwaterloo.ca

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

ABSTRACT

This paper presents a new algorithm for computing the Hermite form of a polynomial matrix. Given a nonsingular $n \times n$ matrix A filled with degree d polynomials with coefficients from a field, the algorithm computes the Hermite form of A using an expected number of $(n^3 d)^{1+o(1)}$ field operations. This is the first algorithm that is both softly linear in the degree d and softly cubic in the dimension n. The algorithm is randomized of the Las Vegas type.

Categories and Subject Descriptors

G.4 [**Mathematical Software**]: Algorithm Design and Analysis; I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms

Keywords

Hermite form, Polynomial matrices

1. INTRODUCTION

Among the classical normal forms for matrices over a principal ideal domain, the Hermite form is the best known. Recall the definition of the form over the ring $\mathsf{K}[x]$ of univariate polynomials over a field K. Corresponding to any nonsingular $A \in \mathsf{K}[x]^{n \times n}$ is a unimodular matrix $U \in \mathsf{K}[x]^{n \times n}$ such that

$$H = UA = \begin{bmatrix} h_1 & \bar{h}_{12} & \cdots & \bar{h}_{1n} \\ & h_2 & \cdots & \bar{h}_{2n} \\ & & \ddots & \vdots \\ & & & h_n \end{bmatrix}$$

is upper triangular, h_j is monic for $1 \le j \le n$, and $\deg \bar{h}_{ij} < \deg h_j$ for $1 \le i < j \le n$. The problem of computing the Hermite form has received a lot of attention. For example, the thesis [5, 18, 15, 27, 6, 19] and ISSAC papers [25, 24, 26, 16] have addressed this topic.

Modulo determinant algorithms [11, 12, 7], see also [4], compute the Hermite form of A working modulo the determinant and require $O\tilde{~}(n^4 d)$ field operations from K. Matrix multiplication can be introduced [11, 24] to reduce the cost to $O\tilde{~}(n^{\omega+1} d)$, where $2 < \omega \le 3$ is the exponent of matrix multiplication. The iterative approach in [17] gives a deterministic $O(n^3 d^2)$ algorithm, achieving a running time that is cubic in n but at the cost of increasing the exponent of d to two. In this paper we give a Las Vegas algorithm to compute H using an expected number of $O\tilde{~}(n^3 d)$ field operations from K. To the best of our knowledge, this is the first algorithm that achieves a running time that is both softly cubic in the matrix dimension n and softly linear in the dimension d.

To put the problem of computing the Hermite form into context, we note that many problems on polynomial matrices now have algorithm that complete in $O\tilde{~}(n^{\omega} d)$ field operations. Examples include the high-order lifting based linear solving, determinant and Smith form algorithms in [20, 21], the fast row reduction algorithm of [9] and minimal approximant basis algorithms in [9, 28]. The techniques in [14] can be adapted to the case of polynomial matrices and achieve algorithms that are subcubic in n for many problems. It is even known that the explicit inverse of A, which has total size $\Omega(n^3 d)$, can be computed in nearly optimal time $O\tilde{~}(n^3 d)$ [13, 23].

The main difficulty to obtain fast algorithms for the Hermite form seems to be the unpredictability and nonuniformity of the degrees of the diagonal entries. The best *a priori* bound for $\deg h_j$ is jd, $1 \le j \le n$. Summing these a priori bounds gives $\sum_{j=1}^{n} jd \in \Theta(n^2 d)$, which overshoots by a factor of n the *a priori* bound $\sum_{j=1}^{n} \deg h_j = \det A \le nd$. For comparison, for the diagonal Smith form $S := \bar{U} A \bar{V} = \mathrm{Diag}(s_1, \ldots, s_n)$ of A, a canonical form under left and right unimodular multiplication, we have the *a priori* bounds $\deg s_j \le (n/(n-j+1))d$; summing these yields $\Theta(nd \log n)$. These good *a priori* bounds for the invariant factors s_j are exploited, for example, in [20, 8] to get improved algorithms for computing the Smith form.

The key to our approach in this paper is to use the Smith form S of A, together with partial information of a right unimodular transform $\bar{V}$, in order to obtain the Hermite form H of A. Our algorithm has two main phases.

The first phase is to compute the degrees of the diagonal entries of H. We show that this can be accomplished via a

ISSAC'11, June 8–11, 2011, San Jose, California, USA.
Copyright 2011 ACM 978-1-4503-0675-1/11/06 ...$10.00.

unimodular matrix triangularization:

$$\left[\begin{array}{c|c} S & \\ \hline V & I_n \end{array}\right] \longrightarrow \left[\begin{array}{c|c} I_n & * \\ \hline & T \end{array}\right] \in \mathsf{K}[x]^{2n\times 2n}. \qquad (1)$$

The matrix V is obtained from $\bar{V}$ by reducing entries in column j modulo s_j, $1 \leq j \leq n$. We show that the submatrix T in (1) will be left equivalent to A and thus, up to associates, has the same diagonal entries as H. When performing the triangularization in (1), we exploit the fact that S is diagonal by keeping offdiagonal entries in the first n columns of the work work matrix reduced modulo the diagonal entry in the same column. Using the upper bound $\sum_{j=1}^{n} \deg s_j \leq nd$, and by avoiding explicit computation of the offdiagonal entries of T and the block above T, we can compute the diagonal entries of T in $O\tilde{}(n^3 d)$ operations from K.

The second phase of our algorithm uses the knowledge of the degrees of the diagonal entries of H to set up a minimal approximant basis problem for recovering H. In particular, the Hermite form H can recovered by computing a left kernel basis in canonical form for the first n columns of the matrix in (1):

$$\left[\begin{array}{c|c} -HVS^{-1} & H \end{array}\right] \left[\begin{array}{c} S \\ \hline V \end{array}\right] = 0.$$

Our main contribution is to show how to transform the kernel computation shown above to an equivalent problem that can be solved in time $O(n^{\omega}\,\mathsf{B}(d))$ using the fast minimal approximant basis algorithm of [9]. Our problem transformation makes use of the partial linearization and reduction of order techniques in [22].

The rest of this paper is organised as follows. Section 2 gives our algorithm for computing the diagonal entries of H. Section 3 gives the algorithm to compute the entire Hermite form given knowledge of the degrees of the diagonal entries. Section 4 makes some concluding remarks. Our cost model is defined below.

Cost model

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 M for polynomial multiplication: let $\mathsf{M} : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{R}_{>0}$ 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. 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 $a = \mathrm{Quo}(a, b)\, b + \mathrm{Rem}(a, b)$ with $\deg \mathrm{Rem}(a, b) < \deg b$. If a and b have degree bounded by d then both of these operations have cost $O(\mathsf{M}(d))$. M is superlinear: $\mathsf{M}(ab) \leq \mathsf{M}(a)\mathsf{M}(b)$ for $a, b \in \mathbb{Z}_{>1}$.

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

$$\left[\begin{array}{cc} s & t \\ u & v \end{array}\right] \left[\begin{array}{c} a \\ b \end{array}\right] = \left[\begin{array}{c} g \\ \; \end{array}\right], \qquad (2)$$

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 assume that $\mathsf{B}(d) = \mathsf{M}(d) \log d$.

2. THE DIAGONAL ENTRIES

Throughout this section let $A \in \mathsf{K}[x]^{n\times n}$ be nonsingular with degree d.

In this section we show how to pass over the Smith form of A in order to recover the degrees of the diagonal entries of the Hermite form of A. The algorithm actually recovers the diagonal entries and not just the degrees, but it is the degrees that will be required by our Hermite form algorithm in the next section. Some mathematical background and previous results are developed and recalled in Subsection 2.1. The algorithm for computing the diagonal entries is given in Subsection 2.2.

2.1 Hermite form via kernel basis

The Hermite form is a canonical form for left equivalence over $\mathsf{K}[x]$. A Hermite form H is *the Hermite form of A* if H is left equivalent to A: $H = UA$ for a unimodular transformation U. Solving for U gives $U = HA^{-1}$. The following lemma gives an alternative, equivalent criteria for a Hermite form H to be left equivalent to A that does not explicitly involve U.

Lemma 1. *A Hermite form H is the Hermite form of A if* $\deg \det H \leq \deg \det A$ *and HA^{-1} is over $\mathsf{K}[x]$.*

To obtain a more compact representation of the matrix A^{-1} in Lemma 1 we will pass over the Smith form. Recall the definition: corresponding to A are unimodular matrices $\bar{U}, \bar{V} \in \mathsf{K}[x]^{n\times n}$ such that $S := \bar{U}A\bar{V} = \mathrm{Diag}(s_1, \ldots, s_n)$ is the Smith canonical form of A, that is, each s_i is monic and $s_i \mid s_{i+1}$ for $1 \leq i \leq n-1$. Solving for A^{-1} gives $A^{-1} = \bar{V}S^{-1}\bar{U}$. Considering Lemma 1, and noting that $\bar{U}$ is unimodular, we may conclude that, for any matrix $H \in \mathsf{K}[x]^{n\times n}$, HA^{-1} is over $\mathsf{K}[x]$ if and only if $H\bar{V}S^{-1}$ is over $\mathsf{K}[x]$. Multiplying S^{-1} by s_n, the largest invariant factor, gives $s_n S^{-1} = \mathrm{Diag}(s_n/s_1, \ldots, s_n/s_n) \in \mathsf{K}[x]^{n\times n}$. We conclude that $H\bar{V}S^{-1}$ is over $\mathsf{K}[x]$ if and only if $H\bar{V}(s_n S^{-1}) \equiv 0 \bmod s_n$. We obtain the following result.

Lemma 2. *Suppose $S = \bar{U}A\bar{V} = \mathrm{Diag}(s_1, \ldots, s_n)$ is the Smith form of A, where $\bar{U}$ and $\bar{V}$ are unimodular, and let $V \in \mathsf{K}[x]^{n\times n}$ be the matrix obtained from $\bar{V}$ by reducing column j of $\bar{V}$ modulo s_j, $1 \leq j \leq n$. Then a Hermite form H is the Hermite form of A if* $\deg \det H \leq \deg \det A$ *and* $H\bar{V}(s_n S^{-1}) \equiv 0 \bmod s_n$.

The following corollary of Lemma 2 is the basis for our approach to compute the diagonal entries of the Hermite form of A.

Corollary 3. *Let V and S be as in Lemma 2. The Hermite form of*

$$\left[\begin{array}{c|c} S & \\ \hline V & I_n \end{array}\right] \in \mathsf{K}[x]^{2n\times 2n} \qquad (3)$$

has the shape

$$\left[\begin{array}{c|c} I_n & * \\ \hline & H \end{array}\right] \in \mathsf{K}[x]^{2n\times 2n}, \qquad (4)$$

where H is the Hermite form of A.

Proof. First note that $\left[\begin{array}{c|c} S & V^T \end{array}\right]^T$ is left equivalent to $\left[\begin{array}{c|c} S & \bar{V}^T \end{array}\right]^T$ where V is a unimodular matrix. It follows that the principal $n \times n$ submatrix of the Hermite form of the matrix in (3) must be I_n. It remains to prove that H is the Hermite form of A. The unimodular transformation that transforms the matrix in (3) to its Hermite form in (4) must have the following shape:

$$\left[\begin{array}{c|c} & * \\ \hline -HVS^{-1} & H \end{array}\right]\left[\begin{array}{c|c} S & \\ \hline V & I_n \end{array}\right] = \left[\begin{array}{c|c} I_n & * \\ \hline & H \end{array}\right].$$

The result follows as the last n rows $[\,-HVS^{-1}|H\,]$ of the transformation matrix are a left kernel basis for $[\,S|V^T\,]^T$. □

Theorem 4. *Let $A \in \mathsf{K}[x]^{n\times n}$ be nonsingular of degree d. If $\#\mathsf{K} \geq 8nd$, matrices S and V as in Lemma 2 can be computed in a Las Vegas fashion with an expected number of $O(n^2\,\mathsf{B}(nd))$ operations from K.*

Proof. First compute a row reduced form R of A using the algorithm of [9], or the deterministic variant in [10]. The Las Vegas algorithm supporting [23, Theorem 28] can now be used to compute V and S from R in the allotted time. □

2.2 The algorithm for diagonal entries

Corresponding to a nonsingular input matrix $A \in \mathsf{K}[x]^{n\times n}$ of degree d, let S and V be as in Lemma 2. Instead of working with the matrix in (3) it will be useful to reverse the columns of V. To this end, let P be the $n \times n$ permutation matrix with ones on the antidiagonal. Note that postmultiplying a matrix by P reverses the order of the columns. Our input matrix has the shape

$$\begin{aligned} G \;&= \left[\begin{array}{c|c} P & \\ \hline & I \end{array}\right]\left[\begin{array}{c|c} S & \\ \hline V & I \end{array}\right]\left[\begin{array}{c|c} P & \\ \hline & I \end{array}\right] \\ &= \left[\begin{array}{cccc|cccc} s_n & & & & & & & \\ & s_{n-1} & & & & & & \\ & & \ddots & & & & & \\ & & & s_1 & & & & \\ \hline * & * & \cdots & * & 1 & & & \\ * & * & \cdots & * & & 1 & & \\ \vdots & \vdots & \cdots & \vdots & & & \ddots & \\ * & * & \cdots & * & & & & 1 \end{array}\right] \quad \in \mathsf{K}[x]^{2n\times 2n}, \end{aligned}$$

and satisfies the following properties:

1. $\mathrm{Diag}(s_1, \ldots, s_n)$ is the Smith form of A and hence satisfies $\sum_{j=1}^{n} \deg s_j = \deg\det A \leq nd$, where $d = \deg A$.
2. Off diagonal entries in column j of G have degree less than the diagonal entry in the same column, $1 \leq j \leq n$.

Our goal is to recover the last n diagonal entries of the Hermite form of G. The standard approach to triangularize G, without any regard to cost or concern for growth of degrees, is to use extended gcd computations and unimodular row operations to zero out entries below the pivot entry in each column.

```
for $j$ from $1$ to $2n-1$ do
    for $i$ from $j+1$ to $2n$ do
        $(g, s, t, u, v) := \mathrm{Gcdex}(G[j,j], G[i,j]);$
        $\begin{bmatrix} G[j,*] \\ G[i,*] \end{bmatrix} := \begin{bmatrix} s & t \\ u & v \end{bmatrix}\begin{bmatrix} G[j,*] \\ G[i,*] \end{bmatrix}$
    od
od
```

Note that, for $j = 1, 2 \ldots, n-1$, the first $n-j$ iterations of the inner loop do nothing since the principal $n \times n$ block of G remains upper triangular during the elimination; omitting these vacuous iterations, the following example shows how the shape of the work matrix changes as in the case $n = 3$:

$$\left[\begin{array}{ccc|ccc} s_3 & & & & & \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & * & * & 1 & & \\ * & * & * & & 1 & \\ * & * & * & & & 1 \end{array}\right] \to \left[\begin{array}{ccc|ccc} * & * & * & * & & \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline & * & * & * & & \\ * & * & * & & 1 & \\ * & * & * & & & 1 \end{array}\right] \to \left[\begin{array}{ccc|ccc} * & * & * & * & * & \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline & * & * & * & & \\ & * & * & * & * & \\ * & * & * & & & 1 \end{array}\right]$$

$$\to \left[\begin{array}{ccc|ccc} * & * & * & * & * & * \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline & * & * & * & & \\ & * & * & * & * & \\ & * & * & * & * & * \end{array}\right] \to \left[\begin{array}{ccc|ccc} * & * & * & * & * & * \\ & * & * & * & & \\ & & s_1 & & & \\ \hline & & * & * & & \\ & * & * & * & * & \\ & * & * & * & * & * \end{array}\right] \to \cdots$$

Note that even after only the first column has been eliminated, the upper triangular structure of the trailing $n \times n$ block has been lost, thus necessitating that j range up to $2n$. Our first refinement of the algorithm is to reverse the order of elimination of entries in the southwest block of G, thus preserving the upper triangularity of the southeast block.

```
for $j$ from $1$ to $n$ do
    for $i$ from $2n$ by $-1$ to $n+1$ do
        $(g, s, t, u, v) := \mathrm{Gcdex}(G[j,j], G[i,j]);$
        $\begin{bmatrix} G[j,*] \\ G[i,*] \end{bmatrix} := \begin{bmatrix} s & t \\ u & v \end{bmatrix}\begin{bmatrix} G[j,*] \\ G[i,*] \end{bmatrix}$
    od
od
```

The following example for $n = 3$ shows how the shape of the shape of the work matrix changes during the first few iterations:

$$\left[\begin{array}{ccc|ccc} s_3 & & & & & \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & * & * & 1 & & \\ * & * & * & & 1 & \\ * & * & * & & & 1 \end{array}\right] \to \left[\begin{array}{ccc|ccc} * & * & * & & & * \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & * & * & 1 & & \\ * & * & * & & 1 & \\ & * & * & & & * \end{array}\right] \to \left[\begin{array}{ccc|ccc} * & * & * & & * & * \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & * & * & 1 & & \\ & * & * & & * & * \\ & * & * & & & * \end{array}\right]$$

$$\to \left[\begin{array}{ccc|ccc} * & * & * & * & * & * \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline & * & * & * & * & * \\ & * & * & & * & * \\ & * & * & & & * \end{array}\right] \to \left[\begin{array}{ccc|ccc} * & * & * & * & * & * \\ & * & * & & & * \\ & & s_1 & & & \\ \hline & * & * & * & * & * \\ & * & * & & * & * \\ & & * & & & * \end{array}\right] \to \cdots$$

Our next two refinements of the triangularization algorithm concern the cost.

Initially, we assume that the offdiagonal entries in G are reduced modulo the diagonal entry in the same column. As the algorithm eliminates entries in column j, we can implicitly perform unimodular row operations to reduce entries in column $j+1, \ldots, n$ modulo the diagonal entry in the same column. In the following example, entries that are kept reduced modulo the diagonal entry in the same column are represented by $\bar{*}$.

$$\left[\begin{array}{ccc|ccc} s_3 & & & & & \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline \bar{*} & \bar{*} & \bar{*} & 1 & & \\ \bar{*} & \bar{*} & \bar{*} & & 1 & \\ \bar{*} & \bar{*} & \bar{*} & & & 1 \end{array}\right] \to \left[\begin{array}{ccc|ccc} * & \bar{*} & \bar{*} & & & * \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & \bar{*} & \bar{*} & 1 & & \\ * & \bar{*} & \bar{*} & & 1 & \\ & \bar{*} & \bar{*} & & & * \end{array}\right] \to \left[\begin{array}{ccc|ccc} * & \bar{*} & \bar{*} & & * & * \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & \bar{*} & \bar{*} & 1 & & \\ & \bar{*} & \bar{*} & & * & * \\ & \bar{*} & \bar{*} & & & * \end{array}\right]$$

$$\to \left[\begin{array}{ccc|ccc} * & \bar{*} & \bar{*} & * & * & * \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline & * & * & * & * & * \\ & \bar{*} & \bar{*} & & * & * \\ & \bar{*} & \bar{*} & & & * \end{array}\right] \to \left[\begin{array}{ccc|ccc} * & * & \bar{*} & * & * & * \\ & * & \bar{*} & & & * \\ & & s_1 & & & \\ \hline & * & * & * & * & * \\ & * & \bar{*} & & * & * \\ & & \bar{*} & & & * \end{array}\right] \to \cdots \qquad (5)$$

The second refinement of the algorithm is to keep the $\bar{*}$ entries reduced modulo the diagonal entry in the same column during the elimination.

```
for $j$ from $1$ to $n$ do
    for $i$ from $2n$ by $-1$ to $n+1$ do
        $(g, s, t, u, v) := \mathrm{Gcdex}(G[j, j], G[i, j])$;
        $G[j, j], G[i, j] := g, 0$;
        $U := \begin{bmatrix} s & t \\ u & v \end{bmatrix}$;
        for $k$ from $j+1$ to $n$ do
            $\begin{bmatrix} G[j,k] \\ G[i,k] \end{bmatrix} := \mathrm{Rem}\left(U \begin{bmatrix} G[j,k] \\ G[i,k] \end{bmatrix}, s_{n-k+1}\right)$
        od;
        for $k$ from $n+1$ to $2n$ do
            $\begin{bmatrix} G[j,k] \\ G[i,k] \end{bmatrix} := U \begin{bmatrix} G[j,k] \\ G[i,k] \end{bmatrix}$
        od
    od
od
```

Notice in (5) that entries in the last n columns of the work matrix G are not kept reduced and can suffer from expression swell. However, our goal is to recover only the trailing n diagonal entries of the last n columns of the triangularization of G. To avoid the cost associated with performing the unimodular row operations on the last n columns of the work matrix, we can exploit the special structure of the work matrix and modify the elimination procedure to only keep track of the the last n diagonals. The following illustrates our point with an example for $n = 3$. Let

$$G = \left[\begin{array}{ccc|ccc} s_3 & & & & & \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & * & * & 1 & & \\ * & * & * & & 1 & \\ a_1 & * & * & & & 1 \end{array}\right].$$

The first elimination step computes the extended gcd of s_3 and a_1, $(g, s, t_1, u, v_1) = \mathrm{Gcdex}(s_3, a_1)$, and updates the work matrix to have the following shape:

$$\left[\begin{array}{ccc|ccc} s & & & & & t_1 \\ & 1 & & & & \\ & & 1 & & & \\ \hline & & & 1 & & \\ & & & & 1 & \\ u & & & & & v_1 \end{array}\right] \left[\begin{array}{ccc|ccc} s_3 & & & & & 0 \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & * & * & 1 & & \\ * & * & * & & 1 & \\ a_1 & * & * & & & 1 \end{array}\right] = \left[\begin{array}{ccc|ccc} g & * & * & & & t_1 \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & * & * & 1 & & \\ a_2 & * & * & & 1 & \\ & * & * & & & v_1 \end{array}\right].$$

Continuing the elimination gives

$$\left[\begin{array}{ccc|ccc} * & * & * & & & t_1 \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline * & * & * & 1 & & \\ a_2 & * & * & & 1 & \\ & * & * & & & v_1 \end{array}\right] \rightarrow \left[\begin{array}{ccc|ccc} * & * & * & & t_2 & * \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline a_3 & * & * & 1 & & \\ & * & * & & v_2 & * \\ & * & * & & & v_1 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{ccc|ccc} * & * & * & t_3 & * & * \\ & s_2 & & & & \\ & & s_1 & & & \\ \hline & * & * & v_3 & * & * \\ & * & * & & v_2 & * \\ & a_4 & * & & & v_1 \end{array}\right] \rightarrow \left[\begin{array}{ccc|ccc} * & * & * & * & * & * \\ & * & * & & & t_4 \\ & & s_1 & & & \\ \hline & * & * & v_3 & * & * \\ & * & * & & v_2 & * \\ & & * & & & v_1 v_4 \end{array}\right] \cdots$$

The key observation is that, although the offdiagonal entries in the last n columns are modified during the elimination, they never affect the last n diagonal entries entries which will depend only on the v_i computed by the calls to Gcdex. Our third refinement of the algorithm is to avoid storing and updating any of the offdiagonal entries in the last n columns of the matrix. Instead, we can keep track of the last n diagonal entries using a vector $D \in \mathsf{K}[x]^{1\times n}$.

```
DiagonalHermite$(S, V)$
Input:  • $S \in \mathsf{K}[x]^{n\times n}$, the Smith form of a
          nonsingular $A \in \mathsf{K}[x]^{n\times n}$ of degree $d$.
        • $V \in \mathsf{K}[x]^{n\times n}$, $\deg \mathrm{Col}(V, j) < \deg s_j$, $1 \le j \le n$.
Output: $D \in \mathbb{Z}^{1\times n}$, the degrees of the last $n$ diagonals
          in the Hermite form of $\left[\begin{array}{c|c} S & \\ \hline V & I \end{array}\right]$.

Let $P$ be equal to $I_n$ with columns reversed.
Intialize $G = \left[\begin{array}{c|c} P & \\ \hline & I \end{array}\right] \left[\begin{array}{c} S \\ \hline V \end{array}\right] P$.
Initialize $D = [1, \ldots, 1]$.
for $j$ from $1$ to $n$ do
    for $i$ from $2n$ by $-1$ to $n+1$ do
        $(g, s, t, u, v) := \mathrm{Gcdex}(G[j, j], G[i, j])$;
        $G[j, j], G[i, j] := g, 0$;
        $U := \begin{bmatrix} s & t \\ u & v \end{bmatrix}$;
        for $k$ from $j+1$ to $n$ do
            $U := \mathrm{Rem}(U, s_{n-k+1})$;
            $\begin{bmatrix} G[j,k] \\ G[i,k] \end{bmatrix} := \mathrm{Rem}\left(U \begin{bmatrix} G[j,k] \\ G[i,k] \end{bmatrix}, s_{n-k+1}\right)$
        od;
        $D[i-n] := D[i-n] \times v$
    od
od;
return $[\deg D[1], \ldots, \deg D[n]]$
```

Figure 1: Algorithm `DiagonalHermite`

For $n = 3$, the following shows the state of G and D during the execution of Algorithm `DiagonalHermite`. Here v_i is the value of v on the i'th call to Gcdex in the above algorithm.

$$\begin{array}{cccccccc} G = \left[\begin{array}{ccc} s_3 & & \\ & s_2 & \\ & & s_1 \\ \hline * & * & * \\ * & * & * \\ * & * & * \end{array}\right] & \rightarrow & \left[\begin{array}{ccc} * & * & * \\ & s_2 & \\ & & s_1 \\ \hline * & * & * \\ * & * & * \\ & * & * \end{array}\right] & \cdots \rightarrow & \left[\begin{array}{ccc} * & * & * \\ & s_2 & \\ & & s_1 \\ \hline & * & * \\ & * & * \\ & * & * \end{array}\right] & \rightarrow & \left[\begin{array}{ccc} * & * & * \\ & * & * \\ & & s_1 \\ \hline & * & * \\ & * & * \\ & & * \end{array}\right] & \cdots \\ D = \ [1, 1, 1] & & [1, 1, v_1] & & [v_3, v_2, v_1] & & [v_3, v_2, v_1 v_4] & \end{array}$$

We now bound the running time of Algorithm `DiagonalHermite`. During the elimination of column j, entries in column j remain bounded in degree by the diagonal entry, a divisor of s_{n-j+1}. Thus, each call to Gcdex is bounded by $\mathsf{B}(\deg s_{n-j+1})$ operations from K. The cost of all n^2 calls to Gcdex is thus bounded by $n \sum_{j=1}^{n} \mathsf{B}(\deg s_j) \le n\mathsf{B}(nd)$, using $\sum_{j=1}^{n} \deg s_j \le nd$.

The cost of applying the transformation U, in each iteration of i, is bounded by $c_1 \sum_{k=1}^{n-j+1} \mathsf{M}(\deg s_k)$ for some constant $c_1 > 0$. For every column j, the total cost of applying transformations is bounded by $nc_1 \sum_{k=1}^{n-j+1} \mathsf{M}(\deg s_k)$. Thus the cost of applying the transformation U, in all iterations, is bounded by

$$nc_1 \sum_{j=1}^{n} \sum_{k=1}^{n-j+1} \mathsf{M}(\deg s_k) \le c_1 n^2\, \mathsf{M}(nd),$$

using the superlinearity of M and that fact that $\sum_{j=1}^{n} \deg s_j \le nd$.

Each entry in D is updated n times and also at any time during the execution of the algorithm $\sum_{i=1}^{n} \deg D[i] \le nd$. This provides a bound for the cost of all updates to D as $O(n\mathsf{M}(nd))$.

We obtain the following result.

THEOREM 5. *Algorithm* `DiagonalHermite` *is correct. The cost of the algorithm is $O(n^2\,\mathsf{M}(nd) + n\,\mathsf{B}(nd))$ operations from* K.

3. FROM DIAGONAL TO HERMITE

We begin by defining some notation. Let $\mathbf{e} = (e_1, \ldots, e_n)$ be a tuple of integers and $u = \left[\begin{array}{ccc} u_1 & \cdots & u_n \end{array}\right] \in \mathsf{K}[x]^{1\times n}$. Following [2], the $\mathbf{e}$-degree of u is equal to $\min_i \deg u_i - e_i$. We define $\mathcal{L}_{\mathbf{e}}(A)$ to be the set of row vectors of $\mathcal{L}(A)$ that have nonpositive $\mathbf{e}$-degree, that is, those vectors u that satisfy $\deg u_i \leq e_i$, $1 \leq i \leq n$.

DEFINITION 6. *Let $L \in \mathsf{K}[x]^{*\times n}$ and $\mathbf{e} = (e_1, \ldots, e_n)$ be a tuple of degree constraints. A matrix $G \in \mathsf{K}[x]^{*\times n}$ is a* genset *of type $\mathbf{e} = (e_1, \ldots, e_n)$ for $\mathcal{L}_{\mathbf{e}}(L)$ if*

- *every row of G has nonpositive $\mathbf{e}$-degree,*
- *$\mathcal{L}_{\mathbf{e}}(G) = \mathcal{L}_{\mathbf{e}}(L)$.*

Note that for some tuples $\mathbf{e}$ we may have $\mathcal{L}(\mathcal{L}_{\mathbf{e}}(A)) \subset \mathcal{L}(A)$. In other words, there may not exist a basis for the lattice $\mathcal{L}(A)$ for which every row in the a basis has degree bounded by $\mathbf{e}$. An obvious example is when $\mathbf{e} = (-1, \ldots, -1)$, in which case $\mathcal{L}(\mathcal{L}_{\mathbf{e}}(A))$ has dimension zero.

3.1 From genset to Hermite form

Let $\mathbf{d} = (d_1, \ldots, d_n)$ be the degrees of the diagonal entries of the Hermite form H of a nonsingular $A \in \mathsf{K}[x]^{n\times n}$. Because H is a basis for $\mathcal{L}(A)$, and each row of H has nonpositive $\mathbf{d}$-degree, we have the following result.

LEMMA 7. *$\mathcal{L}(\mathcal{L}_{\mathbf{d}}(A)) = \mathcal{L}(A)$.*

The following lemma shows how to recover H from a genset $\bar{H}$ of type $\mathbf{d}$ for $\mathcal{L}_{\mathbf{d}}(A)$. The lemma follows as a corollary of Lemma 7 and the following fact regarding H: From among all rows of $\mathcal{L}(A)$ which have first $i-1$ entries zero and entry i nonzero, the i'th row of the Hermite form has i'th entry of minimal degree, $1 \leq i \leq n$.

LEMMA 8. *Suppose $\bar{H} \in \mathsf{K}[x]^{m\times n}$ is a genset of type $\mathbf{d}$ for $\mathcal{L}_{\mathbf{d}}(A)$. Let $L \in \mathsf{K}^{m\times n}$: $\mathrm{Col}(L, j) = \mathrm{Coeff}(\mathrm{Col}(\bar{H}, j), x^{d_j})$, $1 \leq j \leq n$. If $U \in \mathsf{K}^{m\times m}$ is a nonsingular matrix such that UL is in reduced row echelon form, then $U\bar{H}$ will have principal $n \times n$ submatrix equal to H, and last $m - n$ rows zero.*

EXAMPLE 9. *Let $\mathsf{K} = \mathbb{Z}/(7)$, and consider the following Hermite form $H \in \mathsf{K}[x]^{3\times 3}$, together with a genset $\bar{H} \in \mathsf{K}[x]^{5\times 3}$ of type $(1, 3, 2)$ for $\mathcal{L}(H)$.*

$$H = \begin{bmatrix} x & x^2+1 & x+2 \\ & x^3+2x^2 & x+3 \\ & & x^2+2 \end{bmatrix}$$

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

The following shows the leading coefficient matrix L of $\bar{H}$, together with a nonsingular matrix $U \in \mathsf{K}^{5\times 5}$ that transforms L to reduced row echelon form, which due to Lemma 8 will necessarily have principle 3×3 submatrix equal to I_3.

$$\begin{array}{cc} U & L \\ \begin{bmatrix} 2&1&6&0&0 \\ 2&6&0&0&0 \\ 5&5&3&0&0 \\ 6&3&5&1&0 \\ 2&3&2&0&4 \end{bmatrix} & \begin{bmatrix} 4&6&6 \\ 1&4&5 \\ 1&2&3 \\ 3&5&4 \\ 2&0&4 \end{bmatrix} \end{array} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \\ & & \\ & & \end{bmatrix}$$

$U\bar{H}$ is equal to the Hermite form H augmented with two zero rows.

3.2 Hermite form via kernel basis

The quantities defined in this subsection will be used in the remaining subsections. Let $A \in \mathsf{K}[x]^{n\times n}$ be nonsingular, with the following quantities precomputed:

- The Smith form $S = \mathrm{Diag}(s_1, \ldots, s_n) \in \mathsf{K}[x]^{n\times n}$ of A.
- A matrix $V \in \mathsf{K}[x]^{n\times n}$ such that $u \in \mathcal{L}(A)$ if and only if uVS^{-1} is over $\mathsf{K}[x]$. The i'th column of V has entries of degree less than $\deg s_i$.
- The degrees $\mathbf{d} = (d_1, \ldots, d_n)$ of the diagonal entries of the Hermite form H of A.

Let $R := -HVS^{-1} \in \mathsf{K}[x]^{n\times n}$. Then

$$\left[\begin{array}{c|c} R & H \end{array}\right] \left[\begin{array}{c} S \\ \hline V \end{array}\right] = 0. \tag{6}$$

Since column i of V has degree strictly less than $\deg s_i$, we have $\deg R \leq D$ where $D = \max_i d_i - 1$. Let $\mathbf{D} = (D, \ldots, D)$, of length n. The matrix $\left[\begin{array}{c|c} R & H \end{array}\right]$ is a basis (with all rows of nonpositive $(\mathbf{D}, \mathbf{d})$-degree) for the left kernel of $\left[\begin{array}{c|c} S & V^T \end{array}\right]^T$. In fact, by Lemma 8, to recover H it will be sufficient to compute a genset $\left[\begin{array}{c|c} \bar{R} & \bar{H} \end{array}\right]$ of type $(\mathbf{D}, \mathbf{d})$ for $\mathcal{L}_{(\mathbf{D},\mathbf{d})}(\left[\begin{array}{c|c} R & H \end{array}\right])$. The next subsection computes such a genset using fast minimal approximant basis computation.

We remark that the transformation of a canonical form computation to that of a kernel computation is used in [2, 3]. In particular, note that

$$\left[\begin{array}{c|c} U & H \end{array}\right] \left[\begin{array}{c} A \\ \hline I_n \end{array}\right] = 0. \tag{7}$$

The setup in (7) requires no precomputation, and is useful if the unimodular transformation U to achieve the Hermite form is also required. What is important in our approach shown in (6) is the shape of the input problem: we will exploit the fact that S is diagonal, with sum of column degrees in both S and V bounded by nd.

3.3 Hermite via minimal approximant basis

Let $G \in \mathsf{K}[x]^{n\times m}$ and $\mathbf{e}$ be a tuple of nonnegative integers. The entries of $\mathbf{e}$ may be considered to be degree constraints. Recall that an order N minimal approximant basis (or σ-basis [1]) of type $\mathbf{e}$ for G is a nonsingular and row reduced matrix $M \in \mathsf{K}[x]^{n\times n}$ such that $MG \equiv 0 \bmod x^N$. The minimality condition means that the rows of M have $\mathbf{e}$-degrees as small as possible.

LEMMA 10. *Let $M \in \mathsf{K}[x]^{2n\times 2n}$ be an order $N = D + \deg s_n + 1$ minimal approximant basis of type $(\mathbf{D}, \mathbf{d})$ for $\left[\begin{array}{c|c} S & V^T \end{array}\right]^T$. The submatrix of rows of M that have nonpositive $(\mathbf{D}, \mathbf{d})$-degree comprise a basis for $\mathcal{L}(\left[\begin{array}{c|c} R & H \end{array}\right])$.*

Proof. Let $v \in \mathsf{K}[x]^{1\times 2n}$ have nonpositive $(\mathbf{D},\mathbf{d})$-degree. The order N is high enough that $v\left[\begin{array}{c|c} S & V^T \end{array}\right]^T = 0$ if and only if $v\left[\begin{array}{c|c} S & V^T \end{array}\right]^T \equiv 0 \bmod x^N$. □

Example 11. *Consider the matrix H from Example 9. The degrees of the diagonal entries of H are $(1,3,2)$ and thus $D = 2$. The Smith form of H is $\mathrm{diag}(1, 1, x^6 + 2x^5 + 2x^4 + 4x^3)$. Since the first two invariant factors are trivial, we can restrict S to its last entry and V to its last column as the input:*

$$\left[\begin{array}{c} S \\ \hline V \end{array}\right] = \left[\begin{array}{c} x^6 + 2x^5 + 2x^4 + 4x^3 \\ \hline 5x^5 + x^3 + 6x^2 + 6 \\ 6x^5 + 3x^4 + 3x^3 + x \\ 4x^5 + 2x^4 + 2x^3 \end{array}\right]. \tag{8}$$

The following shows an order 9 minimal approximant basis M of type $(2,1,3,2)$ for $\left[\begin{array}{c|c} S & V^T \end{array}\right]^T$, rows permuted to be in nondecreasing $(\mathbf{D},\mathbf{d})$-degree.

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

Exactly the first $n = 3$ rows have nonpositive $(\mathbf{D},\mathbf{d})$-degree. For this example, the northeast block of M is the Hermite form of A up to a row permutation. In general, the northeast block will be a genset of full row rank for $\mathcal{L}_{\mathbf{d}}(H)$.

Using directly the approach of Lemma 10 to recover H is too expensive because the required order $N = D + \deg s_n + 1$ of the minimal approximant basis computation is too high. Indeed, we may have $N \in \Omega(nd)$. The reduction of order technique in [22, Section 2] can be used to reduce the order down to one more than times the maximum of the degree constraints in $(\mathbf{D},\mathbf{d})$. Unfortunately, the largest entry in $\mathbf{d}$ and $\mathbf{D}$ may be $\Omega(nd)$. Before applying the reduction of order technique we apply the partial linearization technique from [22, Section 3] to transform to a new minimal approximant basis problem of type $(\mathbf{D},\mathbf{d}_1)$, with all entries of $\mathbf{d}_1$ bounded by d.

We need to recall some notation from [22]. The norm of a tuple of degree constraints $\mathbf{d}$ is defined to be $\|\mathbf{d}\| = (d_1 + 1) + \cdots + (d_n + 1)$. For $b \geq 0$, let ϕ_b be the function which maps a single degree bound d_i to a sequence of degree bounds, all element of the sequence equal to b except for possibly the last, and such that $\|(d_i)\| = d_i + 1 = \|(\phi_b(d_i))\|$. Let $\mathrm{len}(\phi_b(d_i))$ denote the length of the sequence. For example, we have $\phi_3(10) = 3, 3, 2$ with $\mathrm{len}(\phi_3(10)) = 3$, while $\phi_2(11) = 2, 2, 2, 2$ and $\mathrm{len}(\phi_2(11)) = 4$. Computing a genset of type $(\mathbf{D},\mathbf{d})$ for $\mathcal{L}_{(\mathbf{D},\mathbf{d})}(\left[\begin{array}{c|c} R & H \end{array}\right])$ can be reduced to computing an order N genset of type $\mathbf{d}_1 = (\phi_b(d_1), \ldots, \phi_b(d_n))$. Corresponding to $\mathbf{d}_1$ define the following $\bar{n} \times n$ expansion / compression matrix

$$B := \left[\begin{array}{c|c|c} \begin{array}{c} 1 \\ x^{b+1} \\ \vdots \\ x^{(b+1)\mathrm{len}(\phi_b(d_1))-1} \end{array} & & \\ \hline & \begin{array}{c} 1 \\ x^{b+1} \\ \vdots \\ x^{(b+1)(\mathrm{len}(\phi_b(d_2))-1}) \end{array} & \\ \hline & & \ddots \end{array}\right],$$

where $\bar{n} = \sum_i^n \mathrm{len}(\phi_b(d_i)) = \sum_i^n \lceil (d_i + 1)/(b + 1) \rceil$.

Lemma 12. *Let $b \geq 0$ and define $e_i = \lceil (d_i + 1)/(b + 1) \rceil$, $1 \leq i \leq n$. Let M_1 be an order $N = D + \deg s_n + 1$ minimal approximant basis of type $(\mathbf{D},\mathbf{d}_1)$ for $\left[\begin{array}{c|c} S & (BV)^T \end{array}\right]^T$, where $\mathbf{d}_1 = (\phi_b(d_1), \ldots, \phi_b(d_n))$. If $\left[\begin{array}{c|c} \bar{R}_1 & \bar{H}_1 \end{array}\right]$ is the subset of rows of M_1 which have degree bounded by $(\mathbf{D},\mathbf{d}_1)$, then $\left[\begin{array}{c|c} \bar{R}_1 & \bar{H}_1 B \end{array}\right]$ is a genset of type $(\mathbf{D},\mathbf{d})$ for $\mathcal{L}_{(\mathbf{D},\mathbf{d})}(H)$.*

Furthermore, with the choice $b = d$ the row dimension $\bar{n}$ of BV will satisfy $\bar{n} \in O(n)$.

Example 13. *The problem in Example 11 was to compute a minimal approximant of type $(\mathbf{D},\mathbf{d}) = (2, 1, 3, 2)$ for the 4×1 input matrix shown in (8). Consider setting the linearization parameter b in Lemma 12 as $b = 1$. The expanded problem BV is*

$$\overset{B}{\left[\begin{array}{c|c|c} 1 & & \\ \hline & 1 & \\ & x^2 & \\ \hline & & 1 \\ & & x^2 \end{array}\right]} V = \overset{BV}{\left[\begin{array}{c} 5x^5 + x^3 + 6x^2 + 6 \\ \hline 6x^5 + 3x^4 + 3x^3 + x \\ 6x^7 + 3x^6 + 3x^5 + x^3 \\ \hline 4x^5 + 2x^4 + 2x^3 \\ 4x^7 + 2x^6 + 2x^5 \end{array}\right]}. \tag{9}$$

The degree constraints for the expanded problem are

$$\begin{array}{rcl} (\mathbf{D},\mathbf{d}_1) & = & (2, \phi_1(1), \phi_1(3), \phi_1(2)) \\ & = & (2, 1, 1, 1, 1, 0). \end{array}$$

The following shows an order 9 minimal approximant basis of type $(2, 1, 1, 1, 1, 0)$ for $\left[\begin{array}{c|c} S & (BV)^T \end{array}\right]^T$.

$$M = \left[\begin{array}{c|ccccc} 3x+6 & 0 & 0 & 0 & 2 & 1 \\ x^2+4x+5 & 0 & 0 & x+2 & x+3 & 0 \\ x & x & 1 & 1 & x+2 & 0 \\ \hline 3x+6 & 0 & 0 & 0 & x^2+2 & 0 \\ 0 & 0 & x^2 & 6 & 0 & 0 \\ x^4+5x^3+2x^2+x+4 & 0 & 0 & 5 & x & 0 \end{array}\right]$$

The first 3 rows of M have nonpositive $(\mathbf{D},\mathbf{d}_1)$-degree. Applying the compression matrix to the northwest block of M gives a genset $\bar{H}$ of type $\mathbf{d}$ for $\mathcal{L}_{\mathbf{d}}(H)$:

$$\left[\begin{array}{ccccc} 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & x+2 & x+3 & 0 \\ x & 1 & 1 & x+2 & 0 \end{array}\right] B = \overset{\bar{H}}{\left[\begin{array}{ccc} 0 & 0 & x^2+2 \\ 0 & x^3+2x^2 & x+3 \\ x & x^2+1 & x+2 \end{array}\right]}.$$

Note that in this example $\bar{H}$ has full row rank. We remark that, in general, the genset produced using this expansion/compression technique may have linearly dependent rows.

At this point, we have reduced the problem of computing H to that of computing the rows $\left[\begin{array}{c|c} \bar{R}_1 & \bar{H}_1 \end{array}\right]$ of nonpositive $(\mathbf{D},\mathbf{d}_1)$-degree in an order $N = D + \deg s_1 + 1$ minimal approximant basis of type $(\mathbf{D},\mathbf{d}_1)$, namely

$$\left[\begin{array}{c|c} \bar{R}_1 & \bar{H}_1 \end{array}\right] \left[\begin{array}{c} S \\ \hline BV \end{array}\right] = 0 \bmod x^N.$$

The degree constraints $\mathbf{D} = (D, \ldots, D)$ corresponding the columns of $\bar{R}_1$ may still be too large in general, since $D = \max_i d_i - 1 \in \Omega(nd)$ in the worst case. The key idea now is that $\bar{R}_1$ is not required. Let C be a matrix such that $BV - CS$ has each column of degree bounded by s_i, and consider the transformed input:

$$\left[\begin{array}{c|c} I_n & \\ \hline -C & I \end{array}\right] \left[\begin{array}{c} S \\ \hline BV \end{array}\right] = \left[\begin{array}{c} S \\ \hline E \end{array}\right]. \tag{10}$$

Note that each column in E has degree strictly less than the corresponding diagonal entry in S.

Lemma 14. *Let $\mathbf{D}_1 = (d-1, \ldots, d-1)$, of length n. Let M_2 be an order $N = D + \deg s_n + 1$ minimal approximant basis of type $(\mathbf{D}_1, \mathbf{d}_1)$ for $\left[\begin{array}{c|c} S & E^T \end{array}\right]^T$. Let $\left[\begin{array}{c|c} \bar{R}_2 & \bar{H}_2 \end{array}\right]$ be the submatrix of M_2 comprised of rows that have nonpositive $(\mathbf{D}_1, \mathbf{d}_1)$-degree. Then $\bar{H}_2 B$ is a genset of type $\mathbf{d}$ for $\mathcal{L}_{\mathbf{d}}(H)$.*

Proof. The order N is large enough to ensure that

$$\left[\begin{array}{c|c} \bar{R}_2 & \bar{H}_2 \end{array}\right] \left[\begin{array}{c|c} S & E^T \end{array}\right]^T = 0,$$

and (10) gives that $\left[\begin{array}{c|c} \bar{R}_2 - \bar{H}_2 C & \bar{H}_2 \end{array}\right] \left[\begin{array}{c|c} S & (BV)^T \end{array}\right]^T = 0$, which implies that

$$\left[\begin{array}{c|c} \bar{R}_2 - \bar{H}_2 C & \bar{H}_2 B \end{array}\right] \left[\begin{array}{c} S \\ \hline V \end{array}\right]^T = 0,$$

with all rows in $\bar{H}_2 B$ of nonpositive $\mathbf{d}$-degree. But since V has degrees of entries bounded by the corresponding diagonal entry of S, each row of $\bar{R}_2 - \bar{H}_2$ has nonpositive $\mathbf{D}$ degree. We conclude that $\left[\begin{array}{c|c} \bar{R}_2 - \bar{H}_2 C & \bar{H}_2 \end{array}\right] \subseteq \mathcal{L}_{(\mathbf{D},\mathbf{d})}(\left[\begin{array}{c|c} R & H \end{array}\right])$. The other direction is similar. □

Provided we have chosen the linearization parameter b in Lemma 12 to satisfy $b \in \Theta(d)$ (e.g., $b = d$ will suffice), the final minimal approximant problem in Lemma 14 will have dimension $O(n) \times n$. Note that entries of the compression/expansion matrix B are all powers of x. Thus, the only computation (in terms of field operations) required to construct the input problem in Lemma 14 is to construct E from BV by reducing entries in each column i modulo the diagonal entry in the same column of S, $1 \leq i \leq n$.

Example 15. *The problem in Example 13 was to compute a minimal approximant of type $(\mathbf{D}, \mathbf{d}_1) = (2, 1, 1, 1, 1, 0)$ for the partially linearized 6×1 input matrix B shown in (9). Reducing the last 5 entries modulo the the principal entry we obtain the new input*

$$\left[\begin{array}{c} S \\ \hline E \end{array}\right] = \left[\begin{array}{c} x^6 + 2x^5 + 2x^4 + 4x^3 \\ \hline 5x^5 + x^3 + 6x^2 + 6 \\ 6x^5 + 3x^4 + 3x^3 + x \\ 2x^5 + x^4 + 2x^3 \\ \hline 4x^5 + 2x^4 + 2x^3 \\ 6x^5 + 3x^4 + 3x^3 \end{array}\right]. \tag{11}$$

The following shows the submatrix of an order $N = D + \deg s_n + 1 = 9$ minimal approximant basis of type $(\mathbf{D}_1, \mathbf{d}_1) = (0, 1, 1, 1, 1, 0)$ for $\left[\begin{array}{c|c} S & E^T \end{array}\right]^T$ comprised of rows that have nonpositive $(\mathbf{D}_1, \mathbf{d}_1)$-degree:

$$\left[\begin{array}{c|c} \bar{R}_2 & \bar{H}_2 \end{array}\right] = \left[\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & x & 1 & 2x+5 & 3x+1 & 0 \\ 1 & 0 & 0 & x+2 & x+3 & 0 \end{array}\right].$$

Applying the compression matrix B to $\bar{H}_2$ yields a genset $\bar{H}_2 B$ of type $\mathbf{d}$ for $\mathcal{L}_{\mathbf{d}}(H)$.

At this point (Lemma 14) we have reduced the problem of computing H to that of computing the rows $\left[\begin{array}{c|c} \bar{R}_2 & \bar{H}_2 \end{array}\right]$ of nonpositive $(\mathbf{D}_1, \mathbf{d}_1)$-degree of a minimal approximant basis of order $N = D + \deg s_n + 1$ for an input $\left[\begin{array}{c|c} S & E^T \end{array}\right]^T$. If the partial linearization parameter in Lemma 12 was chosen as $b = d$, then E has dimension $O(n) \times n$, and all degree constrains in $(\mathbf{D}_1, \mathbf{d}_1)$ are bounded by d. Since the sum of the column degrees in $\left[\begin{array}{c|c} S & E^T \end{array}\right]^T$ is bounded by nd, the reduction of order technique in [22, Section 2] can be used to transform to an equivalent problem of dimension $O(n) \times O(n)$ and order only $2d+1$. We refer to [22] for details of the reduction of order technique, and only illustrate the technique here on our running example.

Example 16. *In Example 15 we computed an order 9 minimal approximant basis of type $(\mathbf{D}_1, \mathbf{d}_1) = (0, 1, 1, 1, 1, 0)$ for the 6×1 input $F := \left[\begin{array}{c|c} S & E^T \end{array}\right]^T$ shown in (11). Since the maximum degree constraint is 1, we can instead compute an order $2 \cdot 1 + 1 = 3$ minimal approximant basis $\bar{M}$ of type $(\mathbf{D}_1, \mathbf{d}_1, d-1, d-1) = (0, 1, 1, 1, 1, 0, 0, 0)$ for the following input:*

$$\bar{F} = \left[\begin{array}{c|c|c} F & \mathrm{Quo}(F, x^2) & \mathrm{Quo}(F, x^4) \\ \hline & 1 & \\ & & 1 \end{array}\right] \in \mathsf{K}[x]^{8 \times 3}.$$

Indeed, the submatrix of $\bar{M}$ comprised of rows that have nonpositive $(\mathbf{D}_1, \mathbf{d}_1, 0, 0)$-degree can be written as $\left[\begin{array}{c|c} W & * \end{array}\right]$, where W is the submatrix of an order 9 minimal approximant basis of type $(\mathbf{D}_1, \mathbf{d}_1)$ for F.

We obtain the following theorem.

Theorem 17. *Let $A \in \mathsf{K}[x]^{n \times n}$ be nonsingular of degree d. Assuming $\#\mathsf{K} \geq 8nd$, there exists a Las Vegas probabilistic algorithm that computes the Hermite form H of A using an expected number of $O(n^2\, \mathsf{B}(nd))$ field operations from K.*

Proof. By Theorem 4, the Smith form S of A and corresponding V as described in Subsection 3.2 can be computed in a Las Vegas fashion in the allotted time. By Theorem 5, the degrees of the diagonal entries of H can be computed in the allotted time using Algorithm `DiagonalHermite`. Construct column i of the block E of the input $\left[\begin{array}{c|c} S & E^T \end{array}\right]^T$ to the minimal approximant problem of Lemma 14 by reducing modulo s_i the entries in column i of BV, $1 \leq i \leq n$. Compute the rows of nonpositive degree in the minimal approximant indicate in Lemma 14 by first applying the reduction of order technique from [22, Section 2] to obtain a new problem of dimension $O(n) \times O(n)$ and order $2d+1$, and then apply algorithm `PM-Basis` from [9] in time $O(n^\omega\, \mathsf{B}(d))$ operations from K. Finally, use the approach of Lemma 8 to recover the Hermite form from the genset for $\mathcal{L}_{\mathbf{d}}(H)$. □

4. CONCLUSIONS

We have given a Las Vegas algorithm for computing the Hermite form of a nonsingular $A \in \mathsf{K}[x]^{n \times n}$ using $O\tilde{~}(n^3 d)$ field operations form K. The algorithm has four phases:

1. Compute a row reduced form of A.
2. Compute the Smith form S and the image V of a Smith post-multiplier for A.
3. Compute the diagonal entries of H from S and V.
4. Compute H from S and V and the knowledge of the degrees of the diagonal entries of H

We remark that row reduction algorithm of [9] can accomplish phase 1 using an expected number of $O\tilde{~}(n^\omega d)$ field operations, and we have shown how to apply the fast minimal approximant basis algorithm of [9] to accomplish phase 4 in the same time. $O\tilde{~}(n^\omega d)$ algorithms for phases 2 and 3 may be possible by incorporating blocking into the iterative algorithms currently used, although some additional novel ideas seem to be required for phase 2.

5. REFERENCES

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

[2] B. Beckermann, G. Labahn, and G. Villard. Shifted normal forms of polynomial matrices. In S. Dooley, editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '99*, pages 189—196. ACM Press, New York, 1999.

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

[4] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1996.

[5] P. D. Domich. Three new polynomially-time bounded Hermite normal form algorithms. Master's thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1983.

[6] P. D. Domich. *Residual Methods for Computing Hermite and Smith Normal Forms*. PhD thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1985.

[7] P. D. Domich, R. Kannan, and L. E. Trotter, Jr. Hermite normal form computation using modulo determinant arithmetic. *Mathematics of Operations Research*, 12(1):50–59, 1987.

[8] W. Eberly, M. Giesbrecht, and G. Villard. Computing the determinant and Smith form of an integer matrix. In *Proc. 31st Ann. IEEE Symp. Foundations of Computer Science*, pages 675–685, 2000.

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

[10] S. Gupta, S. Sarkar, A. Storjohann, and J. Valeriote. Triangular x-basis decompositions and derandomization of linear algebra algorithms over $\mathsf{k}[x]$. 10 2010. Submitted for publication.

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

[12] C. S. Iliopoulos. Worst-case complexity bounds on algorithms for computing the canonical structure of finite abelian groups and the Hermite and Smith normal forms of an integer matrix. *SIAM Journal of Computing*, 18(4):658–669, 1989.

[13] C. P. Jeannerod and G. Villard. Essentially optimal computation of the inverse of generic polynomial matrices. *Journal of Complexity*, 21:72–86, 2005.

[14] E. Kaltofen and G. Villard. On the complexity of computing determinants. *Computational Complexity*, 13(3–4):91–130, 2004.

[15] S. E. Labhalla. *Complexité en temps polynomial : calcul d'une réduite d'Hermite, les différentes représentations des nombres réels*. Doctorat d'Etat, Université Cadi Ayyad, Faculté des Sciences Semlalia, Marrakech, 1991.

[16] D. Micciancio and B. Warinschi. A linear space algorithm for computing the Hermite normal form. In B. Mourrain, editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '01*, pages 231—236. ACM Press, New York, 2001.

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

[18] A. Storjohann. Computation of Hermite and Smith normal forms of matrices. Master's thesis, Dept. of Computer Science, University of Waterloo, 1994.

[19] A. Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, Swiss Federal Institute of Technology, ETH–Zurich, 2000.

[20] A. Storjohann. High–order lifting. Extended Abstract. In T. Mora, editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '02*, pages 246–254. ACM Press, New York, 2002.

[21] A. Storjohann. High–order lifting and integrality certification. *Journal of Symbolic Computation*, 36(3–4):613–648, 2003. Extended abstract in [20].

[22] A. Storjohann. Notes on computing minimal approximant bases. In W. Decker, M. Dewar, E. Kaltofen, and S. Watt, editors, *Challenges in Symbolic Computation Software*, number 06271 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2006. http://drops.dagstuhl.de/opus/volltexte/2006/776 [date of citation: 2006-01-01].

[23] A. Storjohann. On the complexity of inverting integer and polynomial matrices. *Computational Complexity*, 2010. Accepted for publication.

[24] A. Storjohann and G. Labahn. Asymptotically fast computation of Hermite normal forms of integer matrices. In Y. N. Lakshman, editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '96*, pages 259–266. ACM Press, New York, 1996.

[25] G. Villard. Computing Popov and Hermite forms of polynomial matrices. In Y. N. Lakshman, editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '96*, pages 251–258. ACM Press, New York, 1996.

[26] U. Vollmer. A note on the Hermite basis computation of large integer matrices. In R. Sendra, editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '03*, pages 255–257. ACM Press, New York, 2003.

[27] C. Wagner. *Normalformberechnung von Matrizen über euklidischen Ringen*. PhD thesis, Universität Karlsruhe, 1998.

[28] W. Zhou and G. Labahn. Efficient computation of order basis. In J. P. May, editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '09*. ACM Press, New York, 2009.