Preconditioning of Rectangular Polynomial Matrices for Efficient Hermite Normal Form Computation

Arne Storjohann and George Labahn
Department of Computer Science
University of Waterloo, Waterloo, Canada
{astorjoh,glabahn}@daisy.uwaterloo.ca

Abstract

We present a Las Vegas probabalistic algorithm for reducing the computation of Hermite normal forms of rectangular polynomial matrices. In particular, the problem of computing the Hermite normal form of a rectangular $m \times n$ matrix (with $m > n$) reduces to that of computing the Hermite normal form of a matrix of size $(n+1) \times n$ having entries of similar coefficient size and degree. The main cost of the reduction is the same as the cost of fraction-free Gaussian elimination of an $m \times n$ polynomial matrix. As an application, the reduction allows for the efficient computation of one-sided GCD's of two matrix polynomials along with the solution of the matrix diophantine equation associated to such a GCD.

1 Introduction

Let A be a matrix in $\mathbf{F}[x]^{m \times n}$, $\mathbf{F}$ a field, with full column rank. The Hermite normal form of A is a matrix H in $\mathbf{F}[x]^{m \times n}$ obtainable from A by unimodular row transformations such that H is upper triangular with all diagonal entries monic and such that in each column off-diagonal entries have degree less than the diagonal entry. A unimodular (invertible over $\mathbf{F}[x]$) matrix $U \in \mathbf{F}[x]^{m \times m}$ that satisfies $UA = H$ is called a pre-multiplier for the Hermite normal form. In general, the Hermite normal form can be defined for matrices over any principal ideal domain (cf. Newman [15]) and was initially introduced in 1851 by Hermite [8] for the case of square integer matrices. The Hermite normal form always exists and is unique.

In this paper we consider the problem of computing the Hermite normal form of a rectangular input matrix $\mathbf{F}[x]^{m \times n}$ where $m > n+1$. Our motivation for studying this problem comes from two areas: symbolic integration and linear systems theory. In the first area, Trager's algorithm for the algebraic case of Risch's decision procedure for determining closed form solutions of integrals with algebraic integrands makes heavy use of the Round Two algorithm for the computation of integral bases for algebraic extension fields (cf. Ford [4], Trager [18]). This algorithm requires many Hermite form reductions of rectangular polynomial matrices, in particular, of polynomial matrices where $m = 2n$ and $m = n^2$. In the second area, a central operation in linear systems theory (cf. Kailath [10]) involves computing minimal representations for linear systems. In the case where the linear system is based on matrix fraction forms, such minimal representations require computing (and removing) one-sided greatest common divisors of matrix polynomials. This can be accomplished by computing the Hermite normal form of a rectangular matrix (cf. Section 2).

The classical method for computing the Hermite normal form H is to directly reduce the $m \times n$ input matrix A by applying a sequence of unimodular row operations (essentially Gaussian elimination over the domain of entries). A unimodular pre-multiplier matrix U that satisfies $UA = H$ can be obtained by recording row operations in a companion matrix (initially the $m \times m$ identity matrix). Computing Hermite normal forms over $\mathbf{F}[x]$ when $\mathbf{F}$ is a field of characteristic zero (e.g. $\mathbf{F} = \mathbf{Q}$, the rational numbers) is known to be especially difficult because of the potential for excessive growth in the intermediate expressions; a direct application of the classical method is hopeless in this case. Aside from the problem of intermediate expression swell, we remark that the coefficients of polynomials appearing in the Hermite normal form can typically be much larger than those in the input matrix. For example, consider the case $\mathbf{F} = \mathbf{Q}$. Let A be an $m \times n$ full column rank input matrix over $\mathbf{Q}[x]$. Assume, without loss of generality, that A has been preconditioned to have all coefficients be integer (i.e. A is over $\mathbb{Z}[x]$) and denote by $||A||$ the largest magnitude integer coefficient appearing in A. Such an integer has *length* $O(\log ||A||)$ bits (the number of bits required to represent the integer in binary). The best known bound on the lengths of individual numerators and denominators of rational number coefficients appearing in the Hermite normal form H of A is $O^{\sim}(n^2 d \log ||A||)$ (cf. Storjohann [17, Theorem 4.6]). This is a factor of $O^{\sim}(n^2 d)$ times as large (number of bits) as the lengths of integers appearing in A. This bound applies also to the lengths of numerators and denominators of rational number coefficients appearing in a candidate for U, a unimodular pre-multipler for the Hermite normal form.

The first to show that computing Hermite normal forms over $\mathbf{Q}[x]$ is in $\mathcal{P}$ (the class of polynomial time algorithms) was Kannan in [13]. A fast parallel algorithm for computing the Hermite normal form and pre-multiplier matrix for a square nonsingular polynomial matrix is given by Kaltofen, Krishnamoorthy and Saunders in [11] and a generalization that works for rectangular input matrices in [12]. We remark that the modulo arithmetic algorithms for matrices over the

To appear in ISSAC'95 proceedings, July 1995, Concordia University, Montreal Quebec, Canada.

integers presented in [3, 7, 9] can be modified to work for input matrices over $\mathbf{F}[x]$ but suffer from excessive coefficient growth when $\mathbf{F} = \mathbf{Q}$. A recent method for computing the Hermite normal form of a polynomial matrix is given by Labhalla, Lombardi and Marlin in [14]. Their approach is to convert the problem to one of triangularizing a large matrix over the coefficient field.

It is important to note that the Hermite normal form algorithms in [3, 9, 11, 13] are initially presented for the special case of square nonsingular input matrices. Hafner and McCurley present in [7] a generalization of the modulo arithmetic approach that works for rectangular matrices but they are not able to directly compute a candidate for a pre-multiplier matrix. To handle the case of rectangular input matrices or the case where a candidate for a unimodular pre-multiplier matrix is desired, the authors of [7, 12, 13] reduce to the square nonsingular case. This can be accomplished as follows. Let U_p be an $m \times m$ unimodular matrix such that U_pA consists of a permutation of the rows of A with the first n rows of U_pA linearly independent. Let A_1 be the $n \times n$ matrix consisting of the first n rows of U_pA and let A_2 consist of the last $(m-n)$ rows. Then the $m \times m$ matrix

$$A_s = \begin{bmatrix} A_1 & 0 \\ A_2 & I_{m-n} \end{bmatrix} \tag{1}$$

obtained by permuting the rows of A and augmenting with I_{m-n} is non-singular. Now find an $m \times m$ unimodular matrix U such that $UA_s = H_s$ is the Hermite normal form of A_s. (Note that since A_s and H_s are nonsingular, U is unique and can be computed to be $U \leftarrow H_sA_s^{-1}$.) Let $(UU_p)A = H$. Then H consists of the first n columns of H_s. Take H to be the Hermite normal form of A. Uniqueness of H_s implies uniqueness of H. This method of embedding the rectangular input matrix into a larger square nonsingular matrix can be quite wasteful, particularly in case where the input matrix has dimensions such as $n^2 \times n$.

In section 4 we present our algorithm for simplifying the computation of Hermite normal forms of rectangular matrices of polynomials. In particular, for an $m \times n$ polynomial matrix A we produce a new *preconditioned* polynomial matrix A^* having the same Hermite normal form as that of A. The matrix A^* has entries approximately the same size as those in A but has the added property that only the first $n+1$ rows contain non-zero entries. For example, for an input matrix $A \in \mathbb{Z}[x]^{m \times n}$ with degrees of entries bounded by d, the matrix A^* produced will have entries polynomials bounded in degree by d with coefficients integers bounded in length by $O(\log ||A|| + \log m + \log d)$ bits. The problem of computing the Hermite normal form of A is thus reduced to the same problem for the first $n+1$ rows of A^*. The latter problem can then be computed using any of the Hermite normal form algorithms in [12, 13, 14]. In all cases the subsequent Hermite normal form computation is done for a significantly smaller problem.

The preconditioning is Las Vegas probabilistic in the sense that it will not produce an incorrect reduction but may fail with arbitrarily small probability. The main cost of the preconditioning is the same as the cost of matrix triangularization via fraction-free Gaussian elimination along with approximately $2mn$ polynomial trial divisions. Fraction-free Gaussian elimination over $\mathbf{Q}[x]$ admits good bounds on the size of intermediate expressions. For an input matrix $A \in \mathbb{Z}[x]^{m \times n}$ with degrees of entries bounded by d, intermediate polynomials occurring during the algorithm will have degrees bounded by nd and coefficients integers bounded in length $O(n(\log ||A|| + \log m + \log d))$ bits. In section 6 we give a detailed cost analysis and show how to employ a homomorphic imaging scheme to achieve a fast, practical implementation.

This paper is organized as follows. Section 2 gives a brief description of how to compute a one-sided greatest common divisors of two matrix polynomials by reducing a rectangular matrix of polynomials to Hermite normal form. This section also describes the linear diophantine equation associated to the gcd and its relation to the pre-multiplier matrix for the normal form computation. Section 3 gives the new preconditioning algorithm with section 4 providing an example of the reduction. The proof of correctness follows in section 5 and a cost analysis in section 6.

2 One-sided GCD's of Matrix Polynomials

Let $A = \begin{bmatrix} P^T & Q^T \end{bmatrix}^T$ have full column rank n where P and Q are matrix polynomials of sizes $m_1 \times n$ and $m_2 \times n$, respectively. Let $H = \begin{bmatrix} G^T & O \end{bmatrix}^T$ be the (unique) Hermite normal form of A with G a matrix polynomial of size $n \times n$ and with U a unimodular pre-multiplier matrix with inverse V such that

$$UA = H \text{ with } VU = I \text{ and } UV = I. \tag{2}$$

We can partition U and V into blocks

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}, V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \tag{3}$$

where U_{11} and U_{12} are of size $n \times m_1$ and $n \times m_2$, respectively, and V_{11} and V_{21} are of size $m_1 \times n$ and $m_2 \times n$, respectively. The partitioning in (3) together with equation (2) gives

$$U_{11}P + U_{12}Q = G \tag{4}$$

with

$$P = V_{11}G, \quad Q = V_{21}G, \quad \text{and} \quad U_{11}V_{11} + U_{12}V_{21} = I.$$

From the above equations it is easy to see that G divides both P and Q and that there are no additional non-trivial (i.e. non-unimodular) right divisors of both PG^{-1} and QG^{-1}. Thus, G is a greatest common right divisor of P and Q. Greatest common left divisors can be obtained in a similar fashion using matrix transposes. We refer to equation (4) as the associated diophantine equation for the matrix gcd computation. Note that the solution to (4) is obtained entirely from the first n rows of the pre-multiplier matrix U.

3 The Rectangular HNF Preconditioner

Let A be an $m \times n$ rank n input matrix over $\mathbf{F}[x]$ having $m > n+1$. An invariant of the lattice $\mathcal{L}(A)$ (the set of all linear combinations over $\mathbf{F}[x]$ of the rows of A) is the quantity $h^*(A,n)$, defined to be the gcd of the determinants of all $n \times n$ minors of A. Algorithm `REDUCE` that follows works by preconditioning the input matrix A with a certain random unimodular matrix U_R. With high probability, the gcd of the determinants of the two $n \times n$ minors of U_RA comprised of rows $[1, 2, \ldots, n]$ and rows $[1, 2, \ldots, n-1, n+1]$ will be equal to $h^*(A,n)$. This is sufficient to guarantee that Hermite(A) = Hermite(A^*) where A^* has first $n+1$ rows those of U_RA and all other rows zero. We say the preconditioning is *correct* in this case. (The idea of preconditioning

an input matrix using random unimodular matrices was first used by Kaltofen, Krishnamoorthy and Saunders [11] in the context of Smith normal form computation to reduce the computation of the gcd of the determinants of many minors to the same computation but for only two randomized minors.)

The remainder of the algorithm attempts to find a construction for a unimodular matrix $U^* \in \mathbf{F}[x]^{m\times m}$ that satisfies $U^*A = A^*$. Such a matrix U^* exists if and only if the random unimodular pre-multiplier matrix U_R gives a correct preconditioning. If the preconditioning is bad, this is detected and the algorithm returns FAIL. To bound the probability of failure by a constant ϵ, where $0 < \epsilon < 1$, we require that $\#\mathbf{F} \geq 2\lceil n^2 d/\epsilon \rceil$. Note that this condition on the cardinality of $\mathbf{F}$ is always met for the important case when $\mathbf{F}$ has characteristic zero. In any case, if $\#\mathbf{F}$ is too small, we can compute over an algebraic extension $\mathbb{K}$ of $\mathbf{F}$ having the required number of elements. In this case, the algorithm will produce a matrix $A^* \in \mathbb{K}[x]^{n\times m}$. The Hermite normal form is an entirely rational form so the Hermite normal form of A (over $\mathbf{F}[x]$) can be found by computing, over $\mathbb{K}[x]$, the Hermite normal form of A^*. The only drawback to computing over an extension field $\mathbb{K}$ is that the optionally returned unimodular matrix U^* may not be over $\mathbf{F}[x]$ and that field operations will be slightly more expensive.

Algorithm: REDUCE
Input: A matrix $A \in \mathbf{F}[x]^{m\times n}$ with full column rank and $m > n+1$.
Constant: An upper bound $0 < \epsilon < 1$ on the probability of failing.
Note: We assume that $\#\mathbf{F} \geq 2\lceil n^2 d/\epsilon \rceil$ where d bounds the degrees of entries in A.
Output: A matrix $A^* \in \mathbf{F}[x]^{m\times n}$ with all zero entries in the last $m-n-1$ rows and such that Hermite(A^*) = Hermite(A). Optionally, a unimodular transformation matrix $U^* \in \mathbf{F}[x]^{m\times m}$ such that $U^*A = A^*$.

(1) [Randomize:]
$C \leftarrow$ a subset of $\mathbf{F}$ with $c = \lceil 2n^2 d/\epsilon \rceil$ elements, no two of which are multiplicative inverses of each other;
Note: If $\mathbf{F}$ has characteristic zero, we may choose $C = \{0, 2, \ldots, c\}$. Otherwise, choose $c-1$ distinct pairs of nonzero elements $\{(a_i, a_i^{-1})\}_{1\leq i\leq c-1}$ from $\mathbf{F}$ and set $C = \{0, a_1, a_2, \ldots, a_{c-1}\}$.
$U_1 \leftarrow$ a unit upper triangular matrix in $\mathbf{F}^{(n-1)\times(n-1)}$ with entries chosen at random from C;
$U_2 \leftarrow$ a matrix in $\mathbf{F}^{(n-1)\times(m-n+1)}$ with entries chosen at random from C;
$\vec{\alpha} \leftarrow$ a row vector in $\mathbf{F}^{1\times(m-n+1)}$ with entries chosen at random from C except for $\vec{\alpha}_1 = 1$;
$\vec{\gamma} \leftarrow$ a row vector in $\mathbf{F}^{1\times(m-n+1)}$ with entries chosen at random from C except for $\vec{\gamma}_2 = 1$;
$U_R \leftarrow \begin{bmatrix} U_1 & U_2 \\ O & \vec{\alpha} \\ O & \vec{\gamma} \\ O & O \end{bmatrix} + \begin{bmatrix} O & \\ & I_{m-n-1} \end{bmatrix}$, an $m\times m$ matrix over $\mathbf{F}$;
Note: U_R is unimodular since $\det(U_R) = 1 - \alpha_2\gamma_1$ which by our choice of C is a nonzero element of $\mathbf{F}$.
$B \leftarrow U_R A$, a matrix in $\mathbf{F}[x]^{m\times n}$ with the same Hermite normal form as A;
Note: $B = \begin{bmatrix} B_1 \\ \vec{b_n} \\ \vec{b_{n+1}} \\ B_2 \end{bmatrix}$ where $\vec{b_n}$ and $\vec{b_{n+1}}$ are row vectors and B_1 and B_2 are of size $(n-1)\times n$ and $(m-n-1)\times n$ respectively.

(2) [Find Annihilators:]
$M_1 \leftarrow \begin{bmatrix} B_1 \\ \vec{b_n} \end{bmatrix}$;
$d_1 \leftarrow \det(M_1)$;
$V' \leftarrow -B_2 M_1^{\text{adj}} P_1$ where $P_1 = \begin{bmatrix} I_{n-1} & O \\ O & \begin{bmatrix} 1 & 0 \end{bmatrix} \end{bmatrix}$;
$V \leftarrow \begin{bmatrix} V' & d_1 I_{m-n-1} \end{bmatrix}$;
$M_2 \leftarrow \begin{bmatrix} B_1 \\ \vec{b_{n+1}} \end{bmatrix}$;
$d_2 \leftarrow \det(M_2)$;
$W' \leftarrow -B_2 M_2^{\text{adj}} P_2$ where $P_2 = \begin{bmatrix} I_{n-1} & O \\ O & \begin{bmatrix} 0 & 1 \end{bmatrix} \end{bmatrix}$;
$W \leftarrow \begin{bmatrix} W' & d_2 I_{m-n-1} \end{bmatrix}$;
Note: V and W are left annihilators of B (i.e. VB and WB are the zero matrix).

(3) [Find probable value for $h^*(A,n)$:]
If both d_1 and d_2 are zero then return FAIL;
$g_n^* \leftarrow \gcd(d_1, d_2)$;

(4) [Check that the preconditioning is correct:]
If g_n^* does not divide all entries of V' and W' then return FAIL;

(5) [Construct A^*:]
$A^* \leftarrow \begin{bmatrix} B_1 \\ \vec{b_n} \\ \vec{b_{n+1}} \\ O \end{bmatrix}$, a matrix in $\mathbf{F}[x]^{m\times n}$ with same Hermite normal form as A;
If U^* is not required then output A^* and terminate, otherwise continue.

(6) [Solve extended Euclidean problem:]
$(a,b) \leftarrow$ a solution to: $ad_1 + bd_2 = g_n^*$;

(7) [Construct unimodular multiplier:]
$U^* \leftarrow \begin{bmatrix} I_{n+1} & 0 \\ \frac{a}{g_n^*}V' + \frac{b}{g_n^*}W' & I_{m-n-1} \end{bmatrix} U_R$;

(8) [Output:] U^* and A^*.

Note that the matrices V and W constructed in step (2) are indeed left annihilators of matrix B of step (1). For example, we have

$$\begin{aligned} VB &= \begin{bmatrix} -B_2 M_1^{\text{adj}} P_1 & d_1 I_{m-n-1} \end{bmatrix} \begin{bmatrix} B_1 \\ \vec{b_n} \\ \vec{b_{n+1}} \\ B_2 \end{bmatrix} \\ &= -B_2 M_1^{\text{adj}} P_1 \begin{bmatrix} B_1 \\ \vec{b_n} \\ \vec{b_{n+1}} \end{bmatrix} + d_1 B_2 \\ &= -B_2 M_1^{\text{adj}} M_1 + d_1 B_2 \\ &= -B_2 d_1 + d_1 B_2 \\ &= O \end{aligned}$$

the $(n-m-1)\times m$ zero matrix. A similar decomposition holds for W.

For clarity, and to simplify the complexity analysis in section 6, we have shown how the construction of annihilators V and W reduces to computing adjoints, determinants and matrix multiplication. In practice, we compute suitable annihilators by triangularizing a single $m \times (n+1)$ matrix using fraction-free Gaussian elimination (cf. Geddes, Czapor and Labahn [5] or the original articles by Bareiss [1, 2]). First, let B' be the matrix $[B|\vec{e}] \in \mathbf{F}^{m\times(n+1)}$ where B is as computed in step (1) and $\vec{e}$ is an $m \times 1$ column vector with all entries 0 except for the n-th entry, which is 1. Next, perform fraction-free Gaussian elimination on B', up to column n, and with row pivoting limited to the first n rows. Record row operations in a companion matrix (initially the $m \times m$ identity) to obtain the pre-multiplier

$$V = \begin{bmatrix} V_1 & O \\ V_2 & d_1 I_{m-n-1} \end{bmatrix}.$$

Then, VB' has entries below the diagonal in the first n columns zero whence the matrix $V = \begin{bmatrix} V_2 & d_1 I_{m-n-1} \end{bmatrix}$ is a left annihilator of B. Continue fraction-free Gaussian elimination for one more column, with row pivoting limited to the first $n+1$ rows, and keep recording row operations in the companion matrix. The last $m-n-1$ rows of the companion matrix are now $W = \begin{bmatrix} W_2 & d_2 I_{m-n-1} \end{bmatrix}$, the second annihilator of B. This procedure may break down if the principle $n \times n$ minor of B is singular (i.e. $d_1 = 0$) in which case a zero pivot will be encountered during fraction-free Gaussian elimination. If this is the case, then we set V to be the $(m-n-1) \times m$ zero matrix. Similarly, if $d_2 = 0$, then we set W to be the $(m-n-1) \times m$ zero matrix.

4 Example of HNF Preconditioning

Let P be the matrix

$$\begin{bmatrix} -4x^2+2x-4 & 14x^2-16x-16 & -3x^2-5x+2 \\ 8x^2+4x+6 & -3x^2+60x+23 & 17x+7 \\ -2x^2+2 & 4x^2+8 & -9x+3 \\ -2x^2 & 6x^2-2x-2 & -2x^2-6x-2 \end{bmatrix}$$

and Q be the matrix

$$\begin{bmatrix} -6x^2+4 & 14x^2-2x+14 & -2x^2-24x+4 \\ -6x^2-4x-2 & -x^2-42x-7 & 2x^2-17x-3 \\ -8x^2-4x-8 & x^2-70x-29 & -14x-8 \\ 8x^2+4x+10 & -3x^2+72x+39 & -x^2+11x+7 \\ 2x^2+2 & -2x^2+16x+6 & 2x^2+3x+3 \end{bmatrix}$$

Using the method of section 2 together with algorithm REDUCE we will find a greatest common right divisor G of P and Q as well as a solution (U_{11}, U_{12}) to the associated diophantine equation $U_{11}P + U_{12}Q = G$ which we may write as

$$\begin{bmatrix} U_{11} & U_{12} \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix} = G. \qquad (5)$$

First we apply algorithm REDUCE to the 9×3 rectangular matrix polynomial $A = \begin{bmatrix} P^T & Q^T \end{bmatrix}^T$. In step (1) of REDUCE we choose

$$\begin{aligned} U_1 &= \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}, \\ U_2 &= \begin{bmatrix} 0 & 3 & 0 & 2 & 0 & 2 & 0 \\ 2 & 0 & 3 & 0 & 2 & 2 & 3 \end{bmatrix}, \\ \vec{\alpha} &= \begin{bmatrix} 1 & 0 & 3 & 0 & 0 & 2 & 0 \end{bmatrix}, \\ \vec{\gamma} &= \begin{bmatrix} 2 & 1 & 2 & 0 & 0 & 3 & 0 \end{bmatrix}, \end{aligned}$$

leading to the randomizing matrix

$$U_R = \begin{bmatrix} 1 & 3 & 0 & 3 & 0 & 2 & 0 & 2 & 0 \\ 0 & 1 & 2 & 0 & 3 & 0 & 2 & 2 & 3 \\ 0 & 0 & 1 & 0 & 3 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 1 & 2 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Let B be the 9×3 matrix consisting of the matrix product $U_R A$. In step (2) we obtain

$$V = \begin{bmatrix} V' & d_1 I_5 \end{bmatrix} \text{ and } W = \begin{bmatrix} W' & d_2 I_5 \end{bmatrix}$$

where $d_1 = -3524x^6 - 14102x^5 - 17462x^4 - 20810x^3 - 20998x^2 + 14788x - 340$ and $d_2 = -16236x^5 - 24174x^4 - 2664x^6 - 20952x^3 - 37998x^2 + 23040x - 612$. The greatest common divisor is given by $g = \gcd(d_1, d_2) = x^4 + 2x^3 + x^2 + 4x - 2$ and this divides every entry of both V' and W' which proves that the preconditioning is correct. As a result, the Hermite normal form of A is the same as that of $A^* = \begin{bmatrix} (A^\#)^T & O \end{bmatrix}^T$ where $A^\#$ is the matrix

$$\begin{bmatrix} 18x^2+14x+30 & 15x^2+218x+111 & -7x^2+16x+25 \\ -8x^2+4x+32 & 37x^2+106x+119 & -70x+32-2x^2 \\ -4x^2+34+8x & 40x^2+128+138x & -59x+29-8x^2 \\ 6x^2+42+12x & 33x^2+159+210x & -39x+33-9x^2 \end{bmatrix},$$

comprised of the first four rows of B. We compute the Hermite normal form of $A^\#$ to be

$$H^\# = \begin{bmatrix} G \\ O \end{bmatrix} = \begin{bmatrix} 1 & 3x+4 & -3/2x+1/2 \\ 0 & x^2+2x-1 & 0 \\ 0 & 0 & x^2+2 \\ 0 & 0 & 0 \end{bmatrix}$$

hence the matrix greatest common right divisor of P and Q is G.

To solve the associated diophantine equation (5) we also require finding a 4×4 unimodular pre-multiplier matrix $U^\#$ along with $H^\#$ that satisfies $U^\# A^\# = H^\#$. The desired solution $\begin{bmatrix} U_{11} & U_{12} \end{bmatrix}$ can then be taken as the first 3 rows of the matrix product

$$U^\# \begin{bmatrix} U_1 & U_2 \\ O & \vec{\alpha} \\ O & \vec{\gamma} \end{bmatrix}.$$

Note that $\left[\begin{array}{cc} U_{11} & U_{12} \end{array}\right]$ is completely determined here from U_R and $U^{\#}$, the result of the smaller Hermite normal form computation, and that the computation of the 9×9 matrix U^* in steps (6) and (7) was not required.

5 Algorithm Correctness

We first show that algorithm REDUCE never returns an incorrect result. First note that by construction in step (5) the matrix A^* has the first $n+1$ rows those of $B = U_R A$ and remaining $m-n-1$ rows zero. To prove that A and A^* have the same Hermite normal form it is sufficient to show that A^* is obtainable from A via premultiplication by a unimodular matrix. This is accomplished by the following lemma.

Lemma 1 *If algorithm* REDUCE *does not return FAIL, then the matrix U^* produced in step (7) is unimodular and satisfies $U^*A = A^*$.*

Proof: By construction, the matrices V and W found in step (2) are $(m-n-1) \times m$ left annihilators of $B = U_R A$. It follows that the matrix

$$\begin{aligned} N &= \frac{a}{g_n^*}\left[\begin{array}{cc} V' & d_1 I \end{array}\right] + \frac{b}{g_n^*}\left[\begin{array}{cc} W' & d_2 I \end{array}\right] \\ &= \left[\begin{array}{cc} \frac{a}{g_n^*}V' + \frac{b}{g_n^*}W' & I \end{array}\right] \end{aligned}$$

is also a left annihilator of B where (a,b) is a solution to $ad_1 + bd_2 = g_n^*$ as found in step (6). Furthermore, since step (4) did not return FAIL, we must have g_n^* a divisor of all entries in V' and W' hence N contains only polynomial entries. By construction in step (1), $\det(U_R)$ is a nonzero constant polynomial in $\mathbf{F}[x]$ hence U_R is unimodular. This shows that U^* as constructed in step (7) is is the product of two unimodular matrices so U^* is unimodular. Finally, note that

$$\begin{aligned} U^*A &= \left[\begin{array}{cc} I_{n+1} & O \\ \frac{a}{g_n^*}V' + \frac{b}{g_n^*}W' & I_{m-n-1} \end{array}\right] U_R A \\ &= A^* \end{aligned}$$

■

The challenge lies in proving that algorithm REDUCE is a correct Las Vegas algorithm. Note that by construction, entries of V' and W' are associates of determinants of $n \times n$ minors of B so that step (4) will not return FAIL if $g_n^* = h^*(B,n)$. Thus, we desire that in step (3) the identity $g_n^* = h^*(B,n)$ holds with high probability so that repetition of the algorithm will almost never be necessary. (Recall that $h^*(B,n)$ is the gcd of the determinants of all $n \times n$ minors of B.)

The following lemma assures us that g_n^* will be correct provided that the entries in U_R do not form a root of a certain polynomial bounded in degree by $2n^2d$. By a result of Schwartz [16], the probability of this happening is less than $2n^2d/\#C$ (i.e. less than ϵ). This approach was inspired by two articles of Kaltofen, Krishnamoorthy and Saunders [11, 12]. In particular, the proof of the following lemma hinges on a key result presented in [12].

Lemma 2 *Let A be a matrix in $\mathbf{F}[x]^{m \times n}$, $m > n+1$, of rank n and with the degrees of entries bounded by d. Then there is a polynomial π in $(2m(n+1) - n(n+3))/2$ variables such that if*

(1) U_R in $\mathbf{F}^{(n+1)\times m}$ has the form

$$U_R = \left[\begin{array}{c|c} U_1 & U_2 \\ \hline \vec{0} & \vec{\alpha} \\ \hline \vec{0} & \vec{\gamma} \end{array}\right]$$

where $U_2 \in \mathbf{F}^{(n-1)\times(m-n+1)}$, U_1 is unit upper triangular in $\mathbf{F}^{(n-1)\times(n-1)}$, and $\vec{\alpha}$ and $\vec{\gamma}$ are row vectors in $\mathbf{F}^{1\times(m-n+1)}$ with $\vec{\alpha} = [1, \alpha_2, \alpha_3, \ldots, \alpha_{m-n+1}]$ and $\vec{\gamma} = [\gamma_1, 1, \gamma_3, \ldots, \gamma_{m-n+1}]$;

(2) d_1 is the determinant of the principal n-th minor of $U_R A$;

(3) d_2 is the determinant of the $n \times n$ minor formed by rows $[1, 2, \ldots, n-1, n+1]$ of $U_R A$,

then $\gcd(d_1, d_2) = h^*(A,n)$, unless the $(2m(n+1) - n(n+3))/2$ entries in U_2, $\vec{\alpha}$, $\vec{\gamma}$ and above the diagonal in U_1 form a root of π. The degree of π is no more than $2n^2d$.

Proof: First consider the case when U_R contains indeterminate entries. In particular, let the entry in the i-th row k-th column of $[U_1|U_2]$ be $\rho_{i,k}$ where $\bar{\rho} = (\rho_{i,k})_{1 \le i \le n-1, i < k \le m}$ is a vector of indeterminates and let $\bar{\alpha} = (\alpha_2, \alpha_2, \ldots, \alpha_{m-n+1})$ and $\bar{\gamma} = (\gamma_1, \gamma_3, \ldots, \gamma_{m-n+1})$. By a result of Kaltofen, Krishnamoorthy and Saunders [12, Lemma 3.6] we must have $d_1 = h^*(A,n)p_1$, where $p_1 \in \mathbf{F}[x, \bar{\rho}, \bar{\alpha}]$ either is an irreducible polynomial in $\mathbf{F}[\bar{\rho}, \bar{\alpha}, x] \setminus \mathbf{F}[x]$ or is 1. Similarly, we must have $d_2 = h^*(A,n)p_2$, where $p_2 \in \mathbf{F}[x, \bar{\rho}, \bar{\gamma}]$ either is an irreducible polynomial in $\mathbf{F}[\bar{\rho}, \bar{\gamma}, x] \setminus \mathbf{F}[x]$ or is 1. Hence, we must have $\gcd(d_1, d_2) = h^*(A,n)$ if p_1 is not an associate of p_2. To show this, it will be sufficient to demonstrate that either p_1 depends on $\bar{\alpha}$ or p_2 depends on $\bar{\gamma}$. Let A_s be the submatrix comprised of the last $m-n+1$ rows of A and let $C_{i,j}$ denote the cofactor of the entry in the i-th row j-th column of the principal n-th minor of $U_R A$. Then, we can express d_1 and d_2 according to their n-th row cofactor expansion

$$\left[\begin{array}{c} d_1 \\ d_2 \end{array}\right] = \left[\begin{array}{ccccc} 1 & \alpha_2 & \alpha_3 & \cdots & \alpha_{m-n+1} \\ \gamma_1 & 1 & \gamma_3 & \cdots & \gamma_{m-n+1} \end{array}\right] A_s \left[\begin{array}{c} C_{n,1} \\ C_{n,2} \\ \vdots \\ C_{n,n} \end{array}\right] \quad (6)$$

$$= \left[\begin{array}{ccccc} 1 & \alpha_2 & \alpha_3 & \cdots & \alpha_{m-n+1} \\ \gamma_1 & 1 & \gamma_3 & \cdots & \gamma_{m-n+1} \end{array}\right] \left[\begin{array}{c} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{m-n+1} \end{array}\right] \quad (7)$$

Now, the $C_{n,*}$ in (6) will be independent of $(\bar{\alpha}, \bar{\gamma})$ since they are associates of determinant of minors of the first $n-1$ rows of $U_R A$. In particular, the polynomials q_* in (7) will depend only on $(\bar{\rho}, x)$ and not on $(\bar{\alpha}, \bar{\gamma})$. Since d_1 and d_2 are nonzero (A has rank n), there must exist a smallest integer i, $1 \le i \le m-n+1$ such that q_i is nonzero. If $i = 1$, then d_2 depends on γ_1; if $i = 2$, then d_1 depends on α_2; if $3 \le i \le m-n$ then d_1 depends on α_i and d_2 depends on γ_i. This shows that $\gcd(d_1, d_2) = h^*(A,n)$ as required. An application of Lemma 3.5 in [11] yields the existence of a $2nd \times 2nd$ determinant Δ, whose entries are coefficients of x of d_1 and d_2, such that for any evaluation $(\bar{\rho}, \bar{\alpha}, \bar{\gamma}) \to (\hat{\rho}, \hat{\alpha}, \hat{\gamma})$ where $(\hat{\rho}, \hat{\alpha}, \hat{\gamma})$ is a corresponding list of field elements that are not a root of Δ, $\gcd(d_1, d_2) = h^*(A,i)$. It remains to establish a degree bound for Δ. Coefficients of x of $U_R A$ are of degree 1 whence coefficients of x of d_1 and d_2 will have total degrees bounded by n. This leads to a bound on the total degree of Δ of $2n^2d$. Finally, set $\pi = \Delta$ to complete the proof. ■

Theorem 1 *Algorithm* REDUCE *is correct and requires repetition with probability less than* ϵ. ■

6 Algorithm Complexity

Let $P(d)$ be the number of field operations required to multiply two degree d polynomials over $\mathbf{F}[x]$ and let $M(t)$ be the number of bit operations required to multiply two t bit integers. In this paper we assume standard polynomial and integer arithmetic: $P(d) = d^2$ and $M(t) = t^2$. We also assume standard matrix multiplication: two $n \times n$ matrices over a ring $\mathbf{R}$ can be multiplied in $O(n^3)$ ring operations.

We first derive a complexity in terms of field operations over $\mathbf{F}$. The computation of matrix B in step (1) is especially simple since U_R contains only constant polynomials. Matrix B is found in $O(mn^2 \cdot d)$ field operations and will have entries bounded in degree by d. The main cost of the algorithm will be computing the annihilators $V = \begin{bmatrix} V' & d_1 I_{n-m-1} \end{bmatrix}$ and $W = \begin{bmatrix} W' & d_2 I_{n-m-1} \end{bmatrix}$ in step (2). Entries of V' and W' are determinants of $n \times n$ minors of B. These have degrees bounded by nd. Since we assume that $\#\mathbf{F} \geq 2\lceil 2n^2 d/\epsilon \rceil > nd$ we can use an evaluation/interpolation scheme to compute V and W as follows. Let $B|_{x=x_i}$ denote the matrix obtained from B by evaluating each polynomial entry at $x = x_i$. Choose a list $(x_i)_{i=0..nd}$ of distinct evaluation points in $\mathbf{F}$ and perform the following steps: (1) find the images $(B|_{x=x_i})_{0 \leq i \leq nd}$ at a cost of $O(mn \cdot nP(d) \log d)$ field operations; (2) find $d_1|_{x=x_i}$ and $M_1^{\text{adj}}|_{x=x_i}$ for $i = 0, \ldots, nd$ at a cost of $O(nd \cdot n^3)$ field operations; (3) find $V'|_{x=x_i} = -B_2|_{x=x_i} M_1^{\text{adj}}|_{x=x_i} P_1$ for $i = 0, \ldots, nd$ at a cost of $O(nd \cdot mn^2)$ field operations; (4) use Chinese remaindering to reconstruct d_1 and the at most $(m-n-1)n$ nonzero degree nd polynomial entries in V' from their images at a cost of $O(nm \cdot P(nd) \log nd)$ field operations. The determinant d_2 and matrix W' are found similarly. Assuming standard polynomial arithmetic, this leads to a cost of $O^{\sim}(n^3 d^2 m)$ field operations for computing the annihilators V and W in step (2). This bounds the cost of the gcd computation in step (6) and the $O(mn)$ trial divisions in step (4). The construction of U^* in step (7) can be accomplished in $O(n^2 m^2 d)$ field operations by using the obvious block decomposition for the matrix multiplication.

Theorem 2 *For some fixed* ϵ, $0 < \epsilon < 1$, *let* $\mathbf{F}$ *be field that satisfies* $\#\mathbf{F} \geq 2\lceil 2n^2 d/\epsilon \rceil$. *There exists a Las Vegas probabalistic algorithm that takes as input a matrix* $A \in \mathbf{F}[x]^{m \times n}$ *with full column rank, degrees of entries bounded by* d *and with* $m > n+1$, *and returns a matrix* $A^* \in \mathbf{F}[x]^{m \times n}$ *with degrees of entries bounded by* d, *last* $m-n-1$ *rows zero, and having the same Hermite normal form as* A. *Optionally, the algorithm returns a unimodular* $U^* \in \mathbf{F}[x]^{m \times m}$ *with degrees bounded by* $2nd$ *that satisfies* $U^* A = A^*$. *The algorithm requires repetition with probability less than* ϵ. *Using standard polynomial and matrix multiplication, the algorithm requires an expected number of* $O^{\sim}(n^3 d^2 m)$ *field operations from* $\mathbf{F}$ *to produce* A^* *alone and an expected number of* $O^{\sim}(n^2 dm(nd+m))$ *field operations from* $\mathbf{F}$ *to produce* A^* *together with* U^*. ■

Now consider the case when $\mathbf{F} = \mathbf{Q}$. Without loss of generality, and as is done in [13], we assume that the input matrix has been preconditioned to have all integer coefficients. Although we are implicitly computing over $\mathbf{Q}[x]$, beginning with an input matrix $A \in \mathbf{Z}[x]^{m \times n}$ allows all computation in steps (1) through (5) of algorithm REDUCE to be accomplished over the simpler domain $\mathbf{Z}[x]$. We start with an input matrix $A \in \mathbf{Z}[x]^{m \times n}$ having degrees of entries bounded by $d-1$. Let $||A||$ denote the largest integer coefficient appearing in A. The integer coefficients appearing in the randomized matrix B computed in step (1) will be only slightly larger than those appearing in A. In particular, we can choose $C = \{0, 2, \ldots, \lceil 2n^2 d/\epsilon \rceil - 1\}$ so that $||B|| = ||U_R A|| \leq m \cdot \lceil 2n^2 d/\epsilon \rceil \cdot ||A||$. In practice, the dominant cost of the algorithm will almost certainly be finding the annihilators V and W of B in step (2). Entries in V and W are determinants of $n \times n$ minors of B. These determinants will be degree (at most) nd polynomials in $\mathbf{Z}[x]$ with integer coefficients bounded in magnitude by $\beta \leq (\sqrt{n}d||B||)^n \leq (\sqrt{n}dm\lceil 2n^2 d/\epsilon \rceil ||A||)^n$. Asymptotically we have $\log \beta = O(n \log md||A||)$. For p a prime, let $A_p = A \bmod p$ be the matrix in $\mathbf{Z}_p[x]^{n \times n}$ obtained from A by replacing each integer coefficient with its image mod p. To compute V and W, we find V_p and W_p over $\mathbf{Z}_p[x]$ for sufficiently many primes p to allow recovery of the integer coefficient appearing in V and W via the Chinese remainder algorithm. To apply the evaluation/interpolation scheme for computing V_p and W_p developed earlier, we need to choose primes p larger than nd to ensure enough evaluation points in the field $\mathbf{Z}_p$. The following lemma from Giesbrecht shows that we can choose all our primes to be $q = \max(6 + \log\log\beta, 1 + \log nd)$ bits in length.

Lemma 3 ([6]) *Let* $x \geq 3$ *and* $l = 6 + \log\log x$. *Then there exist at least* $2\lceil \lceil \log_2(2x) \rceil / (l-1) \rceil$ *primes* p *such that* $2^{l-1} < p < 2^l$. ■

It follows from this lemma that we can choose a list of $s = 2\lceil \lceil (\log 2\beta) \rceil / (q-1) \rceil = \Theta((\log\beta)/q)$ distinct primes $(p_i)_{1 \leq i \leq s}$ that are bounded in length by q bits and that satisfy both $p_i > nd$ for $1 \leq i \leq s$ and $\prod_{1 \leq i \leq s} s_i > \beta$. Next, perform the following steps: (1) find the images $(B_{p_i})_{1 \leq i \leq s}$; (2) find V_{p_i} and d_{p_i} for $i = 1, \ldots, s$ at a cost of $O^{\sim}(s \cdot n^3 d^2 m)$ bit operations using the evaluation/interpolation scheme developed earlier; (3) apply Chinese remaindering to recover the $O(mn^2 d)$ integer coefficients appearing in V and d_1 at a cost of $O(mn^2 d \cdot M(\log\beta) \log s)$ bit operations. Note that the complexity of step (1) will be bounded by that of step (3). Combining these complexity results and assuming standard polynomial and integer multiplication, $P(d) = d^2$ and $M(t) = t^2$, we obtain the following result.

Theorem 3 *Let* $A \in \mathbf{Z}[x]^{m \times n}$ *have full column rank with* $m > n+1$ *and degrees of entries bounded by* d. *The cost of one pass of algorithm* REDUCE *(up to step (5)) with input* A *is* $O^{\sim}(n^4 md(\log||A|| + d)\log||A||)$ *bit operations using standard integer, polynomial and matrix arithmetic plus a single gcd computation and no more than* $O(nm)$ *trial divisions involving polynomials that are factors of entries in the matrices* V *and* W *found in step (2). Entries in* V *and* W *will be polynomials with degrees bounded by* nd *and with integer coefficients bounded in length by* $O(n(\log||A|| + \log m + \log d))$ *bits. If the algorithm does not return FAIL, the matrix* A^* *returned will have entries polynomials bounded in degree by* d *and with integer coefficients bounded in length by* $O(\log||A|| + \log m + \log d)$ *bits.* ■

References

[1] E. H. Bareiss. Sylvester's identity and multistep integer-preserving Gaussian elimination. *Mathematics of Computation*, 22(103):565–578, 1968.

[2] E. H. Bareiss. Computational solution of matrix problems over an integral domain. *Phil. Trans. Roy. Soc. London*, 10:68–104, 1972.

[3] 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, February 1987.

[4] D.J. Ford. *On the Computation of the Maximal Order of a Dedekind Domain.* PhD thesis, Ohio State, 1978.

[5] Keith O. Geddes, S. R. Czapor, and George Labahn. *Algorithms for Computer Algebra.* Kluwer, Bosten, MA, 1992.

[6] Mark Giesbrecht. Fast algorithms for rational forms of integer matrices. In *Proceedings of ISSAC'94*, pages 305–311, Oxford, England, 1994.

[7] James L. Hafner and Kevin S. McCurley. Asymptotically fast triangularization of matrices over rings. *SIAM Journal of Computing*, 20(6):1068–1083, December 1991.

[8] C. Hermite. Sur l'introduction des variables continues dans la théorie des nombres. *J. Reine Angew. Math.*, 41:191–216, 1851.

[9] Costas 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, August 1989.

[10] Thomas Kailath. *Linear Systems.* Prentice Hall, Englewood Cliffs, N.J., 1980.

[11] Erich Kaltofen, M. S. Krishnamoorthy, and B. David Saunders. Fast parallel computation of Hermite and Smith forms of polynomial matrices. *SIAM Journal of Algebraic and Discrete Methods*, 8:683–690, 1987.

[12] Erich Kaltofen, M. S. Krishnamoorthy, and B. David Saunders. Parallel algorithms for matrix normal forms. *Linear Algebra and its Applications*, 136:189–208, 1990.

[13] R. Kannan. Polynomial-time algorithms for solving systems of linear equations over polynomials. *Theoretical Computer Science*, 39:69–88, 1985.

[14] S.E. Labhalla, H. Lombardi, and R. Marlin. Algorithmes de calcul de la réduction d'Hermite d'une matrice à coefficients polynomiaux. In *Comptes-Rendus de MEGA92, Nice, France.* Birkhauser, 1992.

[15] M. Newman. *Integral Matrices.* Academic Press, 1972.

[16] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27:701–717, 1980.

[17] Arne Storjohann. Computation of Hermite and Smith normal forms of matrices. Master's thesis, Dept. of Computer Scicence, University of Waterloo, 1994.

[18] Barry Trager. *Integration of Algebraic Functions.* PhD thesis, Dept. of EECS, M.I.T., 1984.