Probabilistic Computation of the Smith Normal Form of a Sparse Integer Matrix

M. Giesbrecht[†]

Technical Report No. 95/13

Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada, R3T 2N2

December 12, 1995

*Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada, R3T 2N2. Research was supported in part by Natural Sciences and Engineering Research Council of Canada research grant OGP0155376.

Probabilistic Computation of the Smith Normal Form of a Sparse Integer Matrix*

MARK GIESBRECHT†

Abstract

We present a new probabilistic algorithms to compute the Smith normal form of a sparse integer matrix $A \in \mathbb{Z}^{m\times n}$. The algorithm treats A as a "black-box"; A is only used to compute matrix-vector products and we don't access individual entries in A directly. The algorithm requires about $O(m^2 \log \|A\|)$ such black-box evaluations reduced modulo word-sized primes p on vectors in $\mathbb{Z}_p^{n\times 1}$, plus $O(m^2 n \log \|A\|)$ additional bit operations. For sparse matrices this represents a substantial improvement over previously known algorithms. For example, on an $n\times n$ integer matrix A with $O(n \log n)$ non-zero entries, only about $O(n^3 \log^2 \|A\|)$ bit operations are required to find the Smith form. The new algorithm suffers from no "fill-in" or intermediate value explosion, and uses very little additional space. It also admits a simple coarse grain parallelization. The algorithm is probabilistic of the Monte Carlo type – on any input it returns the correct answer with a controlable, exponentially small probability of error.

Introduction

It was proven by Smith (1861) that any $A \in \mathbb{Z}^{m\times n}$ is equivalent to a unique diagonal $S \in \mathbb{Z}^{m\times n}$ under unimodular transformations. That is, there exist unimodular $P \in \mathbb{Z}^{m\times m}$ and $Q \in \mathbb{Z}^{n\times n}$ (i.e., $\det P$, $\det Q = \pm 1$) such that

$$S = PAQ = \begin{pmatrix} s_1 & & & & & \\ & \ddots & & & 0 & \\ & & s_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & 0 & & & & 0 \end{pmatrix} = \operatorname{diag}(s_1, \ldots, s_r, 0, \ldots, 0) \in \mathbb{Z}^{m\times n}$$

where $r = \operatorname{rank}(A)$ and $s_i \mid s_{i+1}$ for $1 \le i \le r-1$. S is called the *Smith normal form* of A and the non-zero diagonal elements of S the *invariant factors* of A.

Computing the Smith normal form of an integer matrix is useful in many applications, including Diophantine analysis (see Newman 1972) and determining the canonical structure of Abelian groups. Recently, algorithms for the Smith normal form have been used to compute the structure of the class group of a number field (see Hafner & McCurley 1989, Buchmann 1988).

In a typical situation we have a finite Abelian group G in which each element $a \in G$ can be written as $g_1^{e_1} g_2^{e_2} \cdots g_m^{e_m}$ with respect to a set of generators $g_1, \ldots, g_m \in G$. We represent a by $(e_1, \ldots, e_m) \in \mathbb{Z}^{n\times 1}$, yielding a homomorphism $\rho : \mathbb{Z}^{m\times 1} \to G$. If $A = [v_1|\cdots|v_n] \in \mathbb{Z}^{m\times n}$, and the lattice $\mathcal{L}$ generated by the columns of A spans the kernel of ρ, then $G \cong \mathbb{Z}^n/\mathcal{L}$. Moreover, if the Smith form of A is $\operatorname{diag}(s_1, \ldots, s_r, 0, \ldots, 0) \in \mathbb{Z}^{m\times n}$, then $G \cong \mathbb{Z}_{s_1} \oplus \cdots \oplus \mathbb{Z}_{s_r}$.

It is often the case that A is sparse (lots of zero entries), and it is desirable to take advantage of this sparsity when computing the Smith form. Existing algorithms do not do this, and suffer

*Research was supported in part by Natural Sciences and Engineering Research Council of Canada research grant OGP0155376.

†Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada, R3T 2N2, phone: (204)474-8838, fax: (204)269-9178, email: mwg@cs.umanitoba.ca

from "fill-in" (much like Gaussian elimination) as well as coefficient growth (see Kannan & Bachem 1979, Chou & Collins 1982, Domich *et al.* 1987, Iliopolous 1989; the problem of coefficient growth is somewhat avoidable – see Havas *et al.* (1993), Giesbrecht (1995), Hafner & McCurley (1991)). It is suggested by Hafner & McCurley (1989) that perhaps the "black-box" methods of Wiedemann (1986) for solving sparse linear systems might be adapted to computing the determinant of a sparse lattice. We show that this is indeed the case, and that in fact the complete Smith form can be computed using an iterative, black-box approach.

For convenience we use "soft-Oh" notation our cost analyses: for any $f, g : \mathbb{R}^l \to \mathbb{R}$, $f = O\tilde{}(g)$ if and only if $f = O(g \cdot \log^c g)$ for some constant $c > 0$. For $v = \in \mathbb{Z}^{n\times 1}$ we write $\|v\| = \|v\|_\infty$ and for $B \in \mathbb{Z}^{m\times n}$ we write $\|B\| = \|B\|_\Delta = \max_{ij} |B_{ij}|$.

In Section 1 we present the basic structure and intuition for our fast new probabilistic algorithm to compute the Smith normal form of A. Throughout we suppose $A \in \mathbb{Z}^{m\times n}$ has rank r and $m \le n$. Recall that the kth determinantal divisor $d_k \in \mathbb{Z}$ of A is the GCD of all $k \times k$ minors of A. The Smith form can be written as $S = \operatorname{diag}(d_1, d_2/d_1, \ldots, d_r/d_{r-1}, 0, \ldots, 0) \in \mathbb{Z}^{m\times n}$. We use this formulation in the algorithms here, and compute the determinantal divisors from which the Smith form is trivially constructed.

The basic algorithm converges on the determinantal divisors $(d_1, \ldots, d_r) \in \mathbb{Z}^r$ of A as a sequence of a logarithmic number l of integer tuples $\bar{d}^{(1)}, \bar{d}^{(2)}, \ldots, \bar{d}^{(l)} \to \bar{d} \in \mathbb{Z}^r$. The jth such tuple is constructed from the minimal polynomial $g^{(j)} \in \mathbb{Z}[x]$ of a linear recurring integer sequences generated from a random perturbation of A. If $g^{(j)}$ has the form $x \cdot \sum_{0\le i\le r} b_i^{(j)} x^i$ (as it usually will), then we set $d_i^{(j)} = \gcd(d_i^{(j-1)}, b_{r-i}^{(j)})$. Convergence is p-adic: for any prime p and any i,

$$\operatorname{ord}_p(d_i^{(1)}) \ge \operatorname{ord}_p(d_i^{(2)}) \ge \cdots \ge \operatorname{ord}_p(d_i^{(l)}) \stackrel{?}{=} \operatorname{ord}_p(d_i),$$

where equality hopefully holds with high probability after the lth iteration. While the basic algorithm works well in practice, we cannot prove that it converges on the determinantal divisors in a reasonable amount of time. A provably convergent variant algorithm is shown in Section 2 at an additional logarithmic factor in the cost.

The cost of our algorithm will be measured in terms of the number of modular matrix-vector products by A required and the number of auxiliary bit operations needed. In particular, there is no fill-in of zero entries. The algorithm we propose will require $O\tilde{}(m^2 \log \|A\| \cdot \log(1/\epsilon))$ modular matrix-vector products ($Av \bmod p$) where $v \in \mathbb{Z}_p^{n\times 1}$ and p is a (single-word) prime with $O(\log m + \log\log \|A\|)$ bits. It will also require an additional $O\tilde{}((m^2 n \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ bit operations and storage for $O\tilde{}(m^2 \log \|A\| + n)$ bits. This analysis assumes standard (quadratic) integer arithmetic. The algorithm is probabilistic of the Monte Carlo type: ϵ is a user specified parameter, and the output of the algorithm is guaranteed correct with probability greater than $1 - \epsilon$.

The algorithm also admits as straightforward course-grain parallelization: on $O(m)$ processors, each processor will be required to execute $O\tilde{}(m \log \|A\| \cdot \log(1/\epsilon))$ modular matrix-vector products as above, and perform $O\tilde{}((mn \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ additional bit operations, a linear speedup in m. Such a parallelization is not obvious for any of the previously known algorithms.

For comparison, consider computing the Smith form of an $A \in \mathbb{Z}^{n\times n}$ with $O(n \log n)$ non-zero entries. The currently best-known (deterministic) algorithm of Storjohann (1995) requires $O\tilde{}(n^4 \log^2 \|A\| + n^3 \log^2 \|A\|)$ bit operations and $O(n^3 \log \|A\|)$ bits of additional memory, and takes little advantage of the sparsity of A (the difficult to analyse heuristic algorithms of Havas *et al.* (1993), which work very well in practice, optimistically have about the same asymptotic complexity as Storjohann's algorithm. Previous algorithms to this which have been rigorously analysed require at least an additional factor of n in their costs). By contrast, the new algorithm proposed here requires $O\tilde{}(n^3 \log^2 \|A\|)$ bit operations and only $O\tilde{}(n^2 \log \|A\|)$ bits of additional memory.

Unfortunately, we are not able to compute the transition matrices to the Smith form within this cost. Because of the generally large sizes of the entries in the transition matrices, one cannot expect to compute them in the order of time our algorithms require. Work is continuing on an extension to the algorithm under which we can evaluate a transition matrix at any vector (without ever returning the dense transition matrix itself). Many applications of Smith normal form computations, such as determining the structure of the class group of a number field or the determinant of a lattice of relations, do not require the transition matrices be computed.

Finally, we note that in modular algorithms to compute the Smith normal form, a good estimate (i.e., a small multiple) is required for d_r, the GCD of all maximal rank minors of A. This is the determinant of the integer lattice generated by the columns of A. In practice this is considerably smaller than a priori estimates provided by the Hadamard bound. Storjohann (1995) has suggested that a few iterations of the algorithm proposed here will give the rank r and a small multiple of d_r at substantially lower cost than other known methods. A deterministic, modular algorithm could then use this modulus to compute the Smith form of A and transition matrices efficiently.

In Section 2 we present a more general algorithm which *provably* converges on the determinantal divisors of A. The cost is within a logarithmic factor of the conjectured "practical" method discussed above, and gives a provably fast algorithm for the Smith form with the complexity as described above. The algorithm involves working in a so-called rough extension ring R of $\mathbb{Z}$. This is a direct sum of a very small number of maximal orders of number fields, each of very small degree over $\mathbb{Z}$. This section is a refinement of work presented in Giesbrecht (1995).

Definitions and Notation.

For integers n and $k \le n$, define $\mathcal{C}_k^n = \{(c_1, \ldots, c_k) \in \mathbb{N}^k : 1 \le c_1 < \cdots < c_k \le n\}$. In a principal ideal ring P, with $B \in \mathsf{P}^{m\times n}$, $\sigma = (b_1, \ldots, b_k) \in \mathcal{C}_k^m$ and $\tau = (c_1, \ldots, c_k) \in \mathcal{C}_k^n$ define the *submatrix* $B\left[\begin{smallmatrix}\sigma\\ \tau\end{smallmatrix}\right]$:

$$B\begin{bmatrix}\sigma\\ \tau\end{bmatrix} = \begin{pmatrix} B_{b_1c_1} & \cdots & B_{b_1c_k}\\ \vdots & & \vdots \\ B_{b_kc_1} & \cdots & B_{b_kc_k}\end{pmatrix} \in \mathsf{P}^{k\times k},$$

and the $(k \times k)$ *minor* $B\binom{\sigma}{\tau} = \det B\left[\begin{smallmatrix}\sigma\\ \tau\end{smallmatrix}\right] \in \mathsf{P}$.

1 A new black-box algorithm for the Smith form

We first present a basic outline of an algorithm `DeterminantalDivisors` to compute the rank r and non-zero determinantal divisors $d_1, \ldots, d_r$ of any $A \in \mathbb{Z}^{m\times n}$. Efficient implementation of this algorithm requires working modulo a set of small primes (see below). While this algorithm appears to work very well in practice, the proof of fast convergence to the determinantal divisors is subject to conjecture. In Section 2 a modified version of this algorithm is presented which is proven to work quickly asymptotically, though implementation is not as practical.

Algorithm: `DeterminantalDivisors`

Input: – a "black box" to evaluate $A \in \mathbb{Z}^{m\times n}$ at any $v \in \mathbb{Z}^{n\times 1}$: $v \to Av \in \mathbb{Z}^{m\times 1}$.
– A tolerance $\epsilon > 0$.

Output: The rank r and determinantal divisors $d_1, \ldots, d_r \in \mathbb{Z}$ of A. The determinantal divisors are correct with probability at least $1 - \epsilon$.

(1) Let $r := 0$; $d_1, \ldots, d_m := 0$; count:=0;
Fix a finite set $\mathcal{H} := \{1, \ldots, n^3 \log n\}$; Construct a black-box to evaluate $\hat{A} \in \mathbb{Z}^{n\times n}$: $\hat{A}$ is A augmented by $n-m$ zero rows;

(2) While count $\leq (\log(1/\epsilon) + 3\log(m) + \log\log \|A\|)$ Do

(3) Choose random $\alpha_1, \ldots, \alpha_{n-1}, \beta_1, \ldots, \beta_{n-1}, \gamma_1, \ldots, \gamma_n \in \mathcal{H}^{n\times 1}$;
Construct a black-box to evaluate Bv for at any vector $v \in \mathbb{Z}^{n\times 1}$, where $B = U\hat{A}LD$ and

$$U = \begin{pmatrix} 1 & \alpha_2 & \cdots & \alpha_{n-1} \\ & 1 & \ddots & \vdots \\ \text{\Large 0} & & \ddots & \alpha_2 \\ & & & 1 \end{pmatrix}, \; L = \begin{pmatrix} 1 & & & \\ \beta_2 & 1 & & \text{\Large 0} \\ \vdots & \ddots & \ddots & \\ \beta_n & \cdots & \beta_2 & 1 \end{pmatrix}, \; D = \begin{pmatrix} \gamma_1 & & \\ & \gamma_2 & \text{\Large 0} \\ \text{\Large 0} & & \ddots \\ & & & \gamma_n \end{pmatrix}.$$

Use the black box for A to evaluate $\hat{A}$ at a vector in $\mathbb{Z}^{n\times 1}$. L and U are unit Toeplitz matrices and pre-multiplying a vector by such a matrix is equivalent to multiplication of polynomials of degree n, which can be done very efficiently.

(4) **Compute the characteristic polynomial of B:**

(4.1) Choose random $u \in \mathbb{Z}^{1\times n}$, $v \in \mathbb{Z}^{n\times 1}$ (see below);

(4.2) Compute the first $2m$ terms of the linear recurring sequence

$$uv,\ uBv,\ uB^2v,\ \ldots,\ uB^{2m-1}v \in \mathbb{Z},$$

using the black box for B and the Berlekamp-Massey algorithm;

(4.3) Compute the minimal polynomial $g = x^s + b_{s-1}x^{s-1} + \cdots + b_1x + b_0 \in \mathbb{Z}[x]$ of this sequence; The characteristic polynomial of B is probably $g \cdot x^{n-s}$;

(5) Let k be such that $b_0 = \cdots = b_{k-1} = 0$ and $b_k \neq 0$;
If $k > 1$ Then goto (3) Else Let $\bar{r} := \deg(g) - 1$;

(6) If $\bar{r} = r$ Then
count:=count+1;
Let $d_i = \gcd(d_i, b_{n-i})$ for $1 \leq i \leq r$;

(7) Else If $\bar{r} > r$ Then
Let $r := \bar{r}$; count:=1;
Let $d_i := b_{n-i}$ for $1 \leq i \leq r$;

End While;

(8) Output $(d_1, \ldots, d_r)$;

End.

The Smith form $S \in \mathbb{Z}^{m\times n}$ of A is trivially reconstructed from the determinantal divisors of $d_1, \ldots, d_r$: $S = \mathrm{diag}(d_1, d_2/d_1, d_3/d_2, \ldots, d_r/d_{r-1}, 0, \ldots, 0)$.

The principle behind this algorithm rests on a new characterization of the determinental divisors of a square matrix $\hat{A} \in \mathbb{Z}^{n\times n}$. Let $\Lambda = \{x_2, \ldots, x_{n-1}, y_2, \ldots, y_{n-1}, z_1, \ldots, z_n\}$ be a set of algebraically independent indeterminants. Let $\mathcal{U}, \mathcal{L} \in \mathbb{Z}[\Lambda]^{n\times n}$ be generic $n \times n$ Toeplitz matrices:

$$\mathcal{U} = \begin{pmatrix} 1 & x_2 & \cdots & x_{n-1} \\ & 1 & \ddots & \vdots \\ & 0 & \ddots & x_2 \\ & & & 1 \end{pmatrix}, \mathcal{L} = \begin{pmatrix} 1 & & & \\ y_2 & 1 & & 0 \\ \vdots & \ddots & \ddots & \\ y_n & \cdots & y_2 & 1 \end{pmatrix}, \mathcal{D} = \begin{pmatrix} z_1 & & \\ & z_2 & 0 \\ 0 & & \ddots \\ & & & z_n \end{pmatrix}. \tag{1.1}$$

Theorem 1.1. *Let $\hat{A} \in \mathbb{Z}^{n\times n}$. Let $\mathcal{U}$, $\mathcal{L}$, and $\mathcal{D}$ be as above. Let $f \in \mathbb{Z}[\Lambda][x]$ be the characteristic polynomial of $\mathcal{U}\hat{A}\mathcal{L}\mathcal{D} \in \mathbb{Z}[\Lambda]^{n\times n}$ and suppose $f = \sum_{0\le i\le n} a_i x^i$, where $a_0, \dots, a_n \in \mathbb{Z}[\Lambda]$. Then the content* $\mathrm{cont}(a_{n-k}) \in \mathbb{Z}$ *of a_{n-k} is $d_k \in \mathbb{Z}$, the kth determinantal divisor of $\hat{A}$.*

Proof. It is easily seen that

$$a_{n-k} = (-1)^k \sum_{\sigma\in\mathcal{C}_k^n} (\mathcal{U}\hat{A}\mathcal{L}\mathcal{D})\binom{\sigma}{\sigma},$$

$(-1)^k$ times the sum of the symmetric $k \times k$ minors of $(\mathcal{U}\hat{A}\mathcal{L}\mathcal{D})$.

Let $\kappa = (1, 2, \dots, k) \in \mathcal{C}_k^n$ and $\sigma, \tau \in \mathcal{C}_k^n$. Suppose $\mathcal{U}\binom{\kappa}{\sigma} \cdot \mathcal{L}\binom{\tau}{\kappa} = s(\sigma, \tau) + t(\sigma, \tau)$, where $s(\sigma, \tau)$ is the lexicographically smallest term under the term variable ordering $x_2 < x_3 < \cdots x_n$ and $y_2 < y_3 < \cdots y_n$. Kaltofen & Saunders (1991) show that $s(\sigma, \tau)$ is unique to σ, τ and, moreover, $s(\sigma, \tau)$ does not occur as a term in $t(\mu, \nu)$ for any $\mu, \nu \in \mathcal{C}_k^n$. Expanding minors using the Binet-Cauchy formula, we obtain

$$\begin{aligned}(-1)^k a_{n-k} &= \sum_{\sigma\in\mathcal{C}_k^n} (\mathcal{U}\hat{A}\mathcal{L}\mathcal{D})\binom{\sigma}{\sigma} = (\mathcal{U}\hat{A}\mathcal{L})\binom{\kappa}{\kappa} \cdot z_1 z_2 \cdots z_k + \sum_{\substack{\sigma\in\mathcal{C}_k^n \\ \sigma\ne\kappa}} (\mathcal{U}\hat{A}\mathcal{L})\binom{\sigma}{\sigma} \cdot \mathcal{D}\binom{\sigma}{\sigma} \\ &= \sum_{\sigma,\tau\in\mathcal{C}_k^n} \hat{A}\binom{\sigma}{\tau} s(\sigma,\tau) \cdot z_1 z_2 \cdots z_k + \sum_{\sigma,\tau\in\mathcal{C}_k^n} \hat{A}\binom{\sigma}{\tau} t(\sigma,\tau) \cdot z_1 z_2 \cdots z_k \\ &\qquad + \sum_{\substack{\sigma\in\mathcal{C}_k^n \\ \sigma\ne\kappa}} \sum_{\tau,\mu\in\mathcal{C}_k^n} \mathcal{U}\binom{\sigma}{\tau} \hat{A}\binom{\tau}{\mu} \mathcal{L}\binom{\mu}{\sigma} \mathcal{D}\binom{\sigma}{\sigma}.\end{aligned}$$

Since d_k is the GCD of all $k\times k$ minors of $\hat{A}$, and the content of each term in the above expansion is a minor of $\hat{A}$, $d_k \mid \mathrm{cont}(a_{n-k})$.

For each $\sigma, \tau \in \mathcal{C}_k^n$, a monomial of the form $s(\sigma, \tau) \cdot z_1 z_2 \cdots z_k$ occurs exactly once in the above expansion, with content $\hat{A}\binom{\sigma}{\tau}$. Thus $\mathrm{cont}(a_{n-k})$ divides the GCD of $\hat{A}\binom{\sigma}{\tau}$ for all $\sigma, \tau \in \mathcal{C}_k^n$, or, equivalently, $\mathrm{cont}(a_{n-k}) \mid d_k$. $\square$

Thus coefficient a_{n-k} of the characteristic polynomial of $\hat{A}$ in `DeterminantalDivisors` is an integer polynomial in the indeterimants $x_2, \dots, x_n, y_2, \dots, y_n, z_1, \dots, z_n$. At each iteration of the main loop we evaluate $a_{n-k}(\alpha_2, \dots, \alpha_n, \beta_2, \dots, \beta_n, \gamma_1, \dots, \gamma_n)$ for $k = 1 \dots n$, at random $\alpha_i, \beta_i, \gamma_i \in \mathcal{H}$. Since a_{n-k} has content d_k, each evaluation will be a multiple of d_k. We hope that we are left with only d_k after taking a few GCD's of such evaluations. While we are not able to prove that this convergence happens with the desired probability for the algorithm described above, we can prove it for a modification described in the next section, at the cost of an additional logarithmic factor in the running time. In practice the algorithm converges very quickly to the correct determinantal divisors.

The method employed in `DeterminantalDivisors` for computing the rank of $\hat{A}$ and characteristic polynomial g of $B = U\hat{A}LD$ is derived from Kaltofen & Saunders (1991).

Lemma 1.2. *Let $\hat{A} \in \mathbb{Z}^{n\times n}$ have rank r. There exists a non-zero $\chi \in \mathbb{Z}[x_2, \ldots, x_n, y_2, \ldots, y_n, z_1, \ldots, z_n]$ of degree at most $8r^2$ such that for any $\alpha_2, \ldots, \alpha_n, \beta_2, \ldots, \beta_n, \gamma_1, \ldots, \gamma_n$ in an extension field K of $\mathbb{Q}$ and*

$$B = \begin{pmatrix} 1 & \alpha_2 & \cdots & \alpha_{n-1} \\ & 1 & \ddots & \vdots \\ & & \ddots & \alpha_2 \\ \text{\huge 0} & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & \text{\huge 0} \\ \beta_2 & 1 & & \\ \vdots & \ddots & \ddots & \\ \beta_n & \cdots & \beta_2 & 1 \end{pmatrix} \begin{pmatrix} \gamma_1 & & \text{\huge 0} \\ & \gamma_2 & \\ \text{\huge 0} & \ddots & \\ & & \gamma_n \end{pmatrix} \in \mathsf{K}^{n\times n},$$

then if $\chi(\alpha_2, \ldots, \alpha_n, \beta_2, \ldots, \beta_n, \gamma_1, \ldots, \gamma_n) \neq 0$ we have that $\min(B) = x\bar{g} \in \mathsf{K}[x]$ and $\mathrm{char}(B) = x^{n-r}\bar{g} \in \mathsf{K}[x]$ with $\bar{g}(0) \neq 0$ and $\deg \bar{g} = r$. Also, $\deg \bar{g} \leq r$ in all cases.

For efficiency (both practical and theoretical), the characteristic polynomial of B is computed modulo a number of small primes, and the integer result is reconstructed by the Chinese remainder theorem.

Lemma 1.3. *Let K be a finite field of characteristic p. Let $\widetilde{B} \in \mathsf{K}^{n\times n}$ have rank r, minimal polynomial $x\tilde{g} \in \mathsf{K}[x]$ and characteristic polynomial $x^{n-r}\tilde{g}$, where $\tilde{g}(0) \neq 0$ and $\deg \tilde{g} = r$. For a given tolerance ϵ_0 and upper bound m for r, we can compute the rank r of $\widetilde{B}$ correctly with probability at least $1 - \epsilon_0$ using $O(m \log(1/\epsilon_0))$ matrix-vector products $v \to \widetilde{B}v$ for $v \in \mathsf{K}^{n\times 1}$, and $O(mn)$ additional operations in K. The calculated rank is always at most the actual rank. If r is given (or the computed r assumed to be correct) we can compute $\tilde{g}$ without error in this same time. The algorithm requires additional memory for $O(n)$ elements of K.*

Proof. Let $\mathcal{H}_{\mathsf{K}}$ be a subset of K. Choose random $u = (u_1, \ldots, u_n) \in \mathcal{H}_{\mathsf{K}}^{1\times n}$, $v = (v_1, \ldots, v_n) \in \mathcal{H}_{\mathsf{K}}^{n\times 1}$ and using the Berlekamp-Massey algorithm compute the minimal polynomial $\tilde{g}_1 \in \mathsf{K}[x]$ of the linearly recurring sequence

$$uv, u\widetilde{B}v, u\widetilde{B}^2v, \ldots, u\widetilde{B}^{2m-1}v, \ldots \in \mathsf{K}$$

(the first $2m$ terms are sufficient to determine this uniquely). Wiedemann (1986) shows that if we repeat this twice, computing $\tilde{g}_1, \tilde{g}_2 \in \mathsf{K}[x]$, the probability $\tilde{g} = \gcd(\tilde{g}_1, \tilde{g}_2)$ is the minimal polynomial of $\widetilde{B}$ is at least 0.30. Since $\deg \tilde{g} \leq \deg \tilde{g}$ the required probabilities follow immediately. The Berlekamp-Massey algorithm on a sequence of length $O(m)$ requires $O(m^2)$ operations in K, while the m inner products in K^n take $O(nm)$ operations in K. $\square$

To compute the characteristic polynomial of B we choose a set of small primes $\mathcal{P} = \{2, 3, 5, \ldots, p_l\}$, where $l = 2m(\log m + \log \|B\|)$. B has rank at most m, so $\mathrm{rank}(B \bmod p) < r = \mathrm{rank}(B)$ for fewer than $m(\log m + \log \|B\|)$ "bad" primes (this is a rather coarse upper bound, which can be substantially improved). Also, $\log |b_i| < m(\log m + \log \|B\|)$ ($b_1, \ldots, b_r$ are the non-zero coefficients in the minimal polynomial $x\bar{g} \in \mathbb{Z}[x]$ of B), so $\mathcal{P}$ contains enough "good" primes to reconstruct g using the Chinese remainder algorithm. Compute the minimal polynomial modulo each prime in $\mathcal{P}$ using Lemma 1.3. Let r be the maximum degree of any of these modular images of $\bar{g}$, and discard any images modulo "bad" primes which have smaller degree. Reconstruct $\bar{g} \in \mathbb{Z}[x]$ using the remaining "good" primes by the Chinese remander algorithm.

Lemma 1.4. *Let $B \in \mathbb{Z}^{n\times n}$ have minimal polynomial $x\bar{g} \in \mathbb{Z}[x]$ and characteristic polynomial $x^{n-r}\bar{g} \in \mathbb{Z}[x]$, where $\bar{g}(0) \neq 0$ and $\deg \bar{g} = r = \operatorname{rank} B$. For a given tolerance ϵ_1 and upper bound m for r, we can compute r correctly with probability at least $1 - \epsilon_1$ with $O\tilde{~}(m^2 \log \|B\| \cdot \log(1/\epsilon))$ evaluations of the black box for B modulo (single word) primes with $O(\log m + \log\log \|B\|)$ bits, and an additional $O\tilde{~}(m^2 n \log \|B\| \cdot \log(1/\epsilon))$ bit operations. If r is given (or the computed r assumed correct), we can compute $\bar{g}$ without error in this same time. The algorithm requires $O\tilde{~}(m^2 \log \|B\| + n)$ bits of additional storage.*

We can now determine the cost of `DeterminantalDivisors`.

Theorem 1.5. *Let $A \in \mathbb{Z}^{m\times n}$ and $\epsilon > 0$. The algorithm* `DeterminantalDivisors` *requires $O\tilde{~}(m^2 \log \|A\| \cdot \log(1/\epsilon))$ modular matrix-vector products ($Av \bmod p$) for $v \in \mathbb{Z}_p^{n\times 1}$ and (single-word) primes p with $O(\log(m) + \log\log \|A\|)$ bits. As well, an additional $O\tilde{~}((m^2 n \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ bit operations and storage for $O\tilde{~}(m^2 \log \|A\| + n)$ bits are required.*

Proof. The main loop does $(\log(1/\epsilon) + 3\log(m) + \log\log \|A\|)$ iterations. In each iteration the main cost is the computation of the characteristic polynomial of B in step (4) and the $r \leq m$ GCD's in steps (6).

It is easily determined that $\log \|B\| = O(\log n + \log \|A\|)$. Since U and L are Toeplitz matrices the cost of a matrix-vector product by U or L is just that of matrix-vector product of degree n polynomials (see, e.g., Kaltofen & Saunders (1991)). Thus, each evaluation of $Bv \bmod p$ for $v \in \mathbb{Z}_p^{n\times 1}$ requires only $O\tilde{~}(n)$ operations in $\mathbb{Z}_p$ in addition to the cost of evaluting $Aw \bmod p$ for some $w \in \mathbb{Z}_p^{n\times 1}$.

The m GCD's involve integers with $O\tilde{~}(m \log \|A\|)$) bits each. Using standard (quadratic) integer arithmetic this requires $O\tilde{~}(m^3 \log^2 \|A\|)$ bit operations. □

2 Finding the content of black-box polynomials

In this section we describe a more general algorithm `FindContents` to identify the contents of a list of multivariate integer polynomials given by a black box. The algorithm provably and quickly finds the contents with controllably small probability of error.

As we saw in the previous section, `DeterminantalDivisors` amounts to finding the contents of a sequence of integer polynomials

$$(f_1(x_1, \ldots, x_s), \ldots, f_r(x_1, \ldots, x_s)) \in \mathbb{Z}[x_1, \ldots, x_s]^r,$$

where $\deg f_i \leq \nu$. In this case f_i is the $(n-i)$th coefficient of the characteristic polynomial of $\mathcal{U}\hat{A}\mathcal{L}\mathcal{D}$ with content d_i, the ith determinantal divisor of $\hat{A}$, $s = 3n - 2$ and $\nu = 3n$. This list of polynomials is given by a black box, that is, we do not have an explicit representation of each polynomial as a linear combination of monomials, but can evaluate $(f_1(a_1, \ldots, a_s), \ldots, f_r(a_1, \ldots, a_s))$ at any point $(a_1, \ldots, a_s)$ with one evaluation of the black box. We aim to find the contents with as few evaluations of our black box as possible, on the "smallest" points possible. `FindContents` works similarly to `DeterminantalDivisisors`, but can be proven to converge quickly to $d_1, \ldots, d_r$ with probability at least $1 - \epsilon$ in $O(\log(1/\epsilon) + \log(m) + \log\log \|A\|)$ iterations. Some of the ideas of this section are developed in Giesbrecht (1995).

Informally, the idea is as follows. We maintain a vector $(c_1, \ldots, c_r) \in \mathbb{Z}^r$ which contains an "approximation" to $(\operatorname{cont}(f_1), \ldots, \operatorname{cont}(f_r))$. Initially we find a point $\vec{a} \in \mathbb{Z}^s$ such that if $(c_1^{(0)}, \ldots, c_r^{(0)}) := (f_1(\vec{a}), \ldots, f_r(\vec{a}))$, then $c_1^{(0)}, \ldots, c_r^{(0)} \neq 0$. If $f_i = d_i g_i$, where $d_i = \operatorname{cont}(f_i) \in \mathbb{Z}$ and $g_i \in \mathbb{Z}[x_1, \ldots, x_s]$ has content 1, then clearly $d_i \mid c_i^{(0)}$ for $1 \leq i \leq r$.

Convergence of the algorithm is measured in terms of the differences in the orders of primes dividing c_i and d_i. Entering iteration $j \geq 1$, we choose a "random" $\vec{a} \in \mathbb{Z}^s$ and evaluate the black box to obtain $(b_1, \ldots, b_r) := f(\vec{a})$. Let

$$(c_1^{(j)}, \ldots, c_r^{(j)}) := (\gcd(b_1, c_1^{(j-1)}), \ldots, \gcd(b_r, c_r^{(j-1)})).$$

Certainly $d_i \mid c_i^{(j)}$ and $c_i^{(j)} \mid c_i^{(j-1)}$ for $1 \leq i \leq r$. Also, for each i and each prime $p \mid d_i$, $\mathrm{ord}_p(c_i^{(j-1)}) \geq \mathrm{ord}_p(c_i^{(j)}) \geq \mathrm{ord}_p(\mathrm{cont}(f_i))$ and $(c_1^{(j)}, \ldots, c_r^{(j)})$ will (hopefully) "converge" on $(d_1, \ldots, d_r)$. Informally this follows since the probability $g_i(\vec{a}) \equiv 0 \bmod p$ is usually low, so the probability $\mathrm{ord}_p(d_i) = \mathrm{ord}_p(c_i^{(j)})$ is quite high. Iterating this processes allows us to make the probability of error arbitrarily small.

Unfortunately, this method cannot be proven to work for small primes p — the order of such a p in c_i cannot be shown to converge to the order of p in d_i when p is smaller than the degree of f_i. The theorem we use to guarantee this convergence — "Schwartz's Lemma" (see Schwartz (1980)) — only implies that $\mathrm{Prob}\{g_i(\vec{a}) \not\equiv 0 \bmod p\} \geq 1 - \deg(g_i)/p$, which is useless for $p < \deg g_i$. A more suitable theorem, which by necessity must use specific properties of the g_i, is not known.

We instead work in a specially constructed "rough" extension ring R of $\mathbb{Z}$. We find a monic, squarefree $\Gamma \in \mathbb{Z}[z]$ of logarithmic degree and set $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$. We construct Γ such that $\Gamma = \Gamma_1\Gamma_2 \cdots \Gamma_l$ where $\Gamma_i \in \mathbb{Z}[z]$ are monic and irreducible, so

$$\mathsf{R} = \mathsf{O}_1 \oplus \mathsf{O}_2 \oplus \cdots \oplus \mathsf{O}_k$$

where $\mathsf{O}_i = \mathbb{Z}[z]/(\Gamma_i)$ is a maximal order in the number field $\mathbb{Q}[z]/(\Gamma_i)$. For each small p dividing c_i we ensure there exists a k such that $\mathsf{O}_k \bmod p$ contains a sufficiently large finite field. Equivalently, $\Gamma_k \bmod p$ has a sufficiently large irreducible factor in $\mathbb{Z}_p[x]$. This ensures that the probability that $g_i(\vec{a}) \not\equiv 0 \bmod p$ is high. See Lemma 2.2 below. That such an extension can be built efficiently is shown in Theorem 2.1 below.

Consider, for example, finding the content of the $f(x) = 2x^2 - 2x \in \mathbb{Z}[x]$. It is easily shown that $4 \mid f(a)$ for all $a \in \mathbb{Z}$, so we cannot identify the content by computing GCD's of evaluations of f at integers. However, in $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$, where $\Gamma = z^2 + z + 1 \in \mathbb{Z}[z]$, we can choose $a = (z + 2 \bmod \Gamma)$, so $f(a) = (4z + 2 \bmod \Gamma) \in \mathsf{R}$. GCD's are not necessarily well defined in R, so we take GCD's of the *contents* of the evaluations at points in R treated as polynomials (reduced modulo Γ) in $\mathbb{Z}[x]$. In this rather trivial case, the content of $4z + 2$ is 2, as is the content of f.

In Giesbrecht (1995) the following theorem on the construction of such rough extensions of $\mathbb{Z}$ is shown:

Theorem 2.1. *There is an algorithm* `BuildRoughExtension` *which takes as input $\eta \in \mathbb{Z}$ and primes $2 \leq p_1, \ldots, p_\kappa \leq \tau$ and as output produces a squarefree, monic $\Gamma \in \mathbb{Z}[z]$ such that for each i, $\Gamma \bmod p_i$ has an irreducible factor in $\mathbb{Z}_{p_i}[z]$ of degree greater than η. The algorithm is probabilistic of the Las Vegas type (it always produces the correct answer) and requires an expected number of $O((\eta^3 + \eta^2 \log \tau) \cdot \kappa \log^2 \tau \log \kappa)$ bit operations. The output $\Gamma \in \mathbb{Z}[z]$ has degree at most $2\eta(4 + 6 \log \kappa) = O(\eta \log \kappa)$ and $\|\Gamma\| \leq (4\eta^2\tau)^{\log \kappa}$.*

The probability of correctly finding the content

We return to the problem of obtaining the content $d_i \in \mathbb{Z}$ of $f_i \in \mathbb{Z}[x_1, \ldots, x_s]$ where $\deg f_i \leq \nu$, for $1 \leq i \leq r$. As discussed above we choose points randomly from a subset $\mathcal{W}$ of a rough extension ring R of $\mathbb{Z}$ at which to evaluate $f_1, \ldots, f_r$. Let $\Gamma \in \mathbb{Z}[z]$ be monic and squarefree of degree $\gamma > 1$ and $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$. Let $\beta \geq \max(6, \nu^2)$ be an integer, and fix

$$\mathcal{H} = \{0, \ldots, \beta - 1\} \subseteq \mathbb{Z}, \qquad \mathcal{W} = \{h \bmod \Gamma : h \in \mathcal{H}[z], \deg h < 5 \log \nu\} \subseteq \mathsf{R}.$$

We will further specify our choices of Γ, γ, and β in the sequel.

To determine the order of a prime p in d_i, assume $f_i = d_i g_i$ for some $g_i \in \mathbb{Z}[x_1, \ldots, x_s]$ of content 1. For $\vec{a} \in \mathsf{R}^s$, $b_i := f_i(\vec{a}) = d_i g_i(\vec{a}) \in \mathsf{R}$. Define $\overline{\mathrm{cont}}(b_i) = \mathrm{cont}(\bar{b}_i)$, where $\bar{b}_i \in \mathbb{Z}[z]$, $\deg \bar{b}_i < \deg \Gamma$ and $\bar{b}_i \equiv b_i \bmod \Gamma$. We have $\overline{\mathrm{cont}}(b_i) = d_i \cdot \overline{\mathrm{cont}}(g_i(\vec{a}))$ so $\mathrm{ord}_p(d_i) \le \mathrm{ord}_p(\overline{\mathrm{cont}}(b_i))$, with equality when $g_i(\vec{a}) \not\equiv 0 \bmod p$.

The next lemma, from Giesbrecht (1995), examines the probability for a $g \in \mathbb{Z}[x_1, \ldots, x_s]$ with content 1, a prime p, and a randomly and uniformly selected point $\vec{a} \in \mathcal{W}^s$, that $g(\vec{a}) \equiv 0 \bmod p$. Two difficulties must be overcome. First, we make random choices uniformly from $\mathcal{W}$, but these are not generally uniform choices from $\mathcal{W} \bmod p$. Second, Corollary 1 of Schwartz (1980) is not useful for small primes ($\le \deg g$), and we must employ the properties of a rough extension ring R constructed with `BuildRoughExtension`. In the following we make use of the fact that R is a rough extension ring. This ensures that in the direct sum of $\mathsf{R} \cong \mathsf{O}_1 \oplus \cdots \oplus \mathsf{O}_l$ as a direct sum of maximal orders, for each selected prime p there exists an j such that $\mathsf{O}_j/(p)$ contains a fairly large extension of $\mathbb{Z}_p$.

Lemma 2.2 (Giesbrecht 1995). *Let p be a prime such that if $p \le \beta$ then $\Gamma \bmod p$ has an irreducible factor in $\mathbb{Z}_p[z]$ of degree at least $5 \log \nu$. Let $g \in \mathbb{Z}[x_1, \ldots, x_s]$ with $\deg g \le \nu$ and $\mathrm{cont}(g) = 1$. Suppose $\vec{a}$ is chosen randomly and uniformly from $\mathcal{W}^s$. Then $\mathrm{Prob}\,\{g(\vec{a}) \equiv 0 \bmod p\} \le 1/\nu$.*

Define $R = \mathbb{Q} \otimes \mathsf{R} = \mathbb{Q}[z]/(\Gamma) \supseteq \mathsf{R}$, with units $R^* = \{a \bmod \Gamma : a \in \mathbb{Q}[z], \gcd(a, \Gamma) = 1\}$. We allow that our black box may not work when $\chi(\vec{a}) \notin R^*$. The following lemma demonstrates this seldom happens.

Lemma 2.3 (Giesbrecht 1995). *Let $\chi \in \mathbb{Z}[x_1, \ldots, x_s] \backslash \{0\}$. For a randomly and uniformly chosen $\vec{a} \in \mathcal{W}^s$, $\mathrm{Prob}\{\chi(\vec{a}) \notin R^*\} \le \deg(\chi) \cdot \deg(\Gamma)/\beta$.*

Algorithm: `FindContents`

Input: $r \ge 2$, $\epsilon > 0$ and a black box which on input $\vec{a} \in \mathsf{R}^s$ evaluates $(f_1(\vec{a}), \ldots, f_r(\vec{a})) \in \mathsf{R}^r$, where $f_1, \ldots, f_r \in \mathbb{Z}[x_1, \ldots, x_s]$, $\deg f_i \le \nu$, and R an extension ring of $\mathbb{Z}$; We allow for the existence of a $\chi \in \mathbb{Z}[x_1, \ldots, x_s] \backslash \{0\}$ of degree $O(\nu^2)$ such that if $\chi(\vec{a}) \notin R^*$ the black box may report "failure" and is not evaluated.

Output: $(\mathrm{cont}(f_1), \ldots, \mathrm{cont}(f_r)) \in \mathbb{Z}^r$, correct with probability $\ge 1 - \epsilon$;

(1) Let $\beta := \max(2(r\nu + \deg \chi), 1000\nu \cdot \deg(\chi) \log(\deg(\chi)), \nu^2, 6)$;

(2) Choose a random $\vec{a} \in \mathcal{H} = \{0, \ldots, \beta - 1\}^s$; let $(c_1^{(0)}, \ldots, c_r^{(0)}) := (f_1(\vec{a}), \ldots, f_r(\vec{a})) \in \mathbb{Z}^r$; if any of $c_1^{(0)}, \ldots, c_r^{(0)} = 0$, repeat (2);

(3) Find $p_1, \ldots, p_\kappa \in \mathbb{N}$, all the primes $< \beta$ which divide $\mathrm{lcm}(c_1^{(0)}, \ldots, c_r^{(0)})$;

(4) Using `BuildRoughExtension` on $p_1, \ldots, p_\kappa$ and $\eta = 5 \log \nu$ construct $\Gamma \in \mathbb{Z}[z]$, such that for $1 \le i \le \kappa$, $\Gamma \bmod p_i$ has an irreducible factor of degree greater than η in $\mathbb{Z}_p[z]$. Let $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$ and $\mathcal{W} = \{h \bmod \Gamma : h \in \mathcal{H}[z], \deg h < \eta\} \subseteq \mathsf{R}$.

(5) For $i := 1$ to $l = \log(1/\epsilon) + \log r + \log\log \max_i |c_i^{(0)}|$ do

(6) Choose random $\vec{a} \in \mathcal{W}^s$;

(7) Let $(b_1, \ldots, b_r) := (f_1(\vec{a}), \ldots, f_r(\vec{a})) \in \mathsf{R}^r$;
If the black box fails to find $(b_1, \ldots, b_r)$ on $\vec{a}$, so $\chi(\vec{a}) \notin R^*$, this choice of $\vec{a}$ may be ignored, and execution continued with next i at step (5);

(8) Let $(\bar{b}_1, \ldots, \bar{b}_r) := (\overline{\mathrm{cont}}(b_1), \ldots, \overline{\mathrm{cont}}(b_r)) \in \mathbb{Z}^r$;

(9) Let $(c_1^{(i)}, \ldots, c_r^{(i)}) := (\gcd(c_1^{(i-1)}, \bar{b}_1), \ldots, \gcd(c_r^{(i-1)}, \bar{b}_r)) \in \mathbb{Z}^r$;
End For;
(10) Output $(c_1^{(l)}, \ldots, c_r^{(l)})$;

Theorem 2.4. *The algorithm* `FindContents` *works correctly as described and produces the correct answer with probability at least* $1-\epsilon$. *An expected number of* $O((\log(1/\epsilon) + \log r + \log\log s + \log\log \max_i \|f_i\|)/\log \nu)$ *evaluations of the black box are needed. These evaluations are in the ring* $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$, *where* $\deg \Gamma = O(\log(\nu)\log(r))$ *and* $\log \|\Gamma\| = O((\log r + \log \nu)^2)$. *The arguments to the black box from* R^s *have height* $(r\nu)^{O(1)}$.

`FindContents` *requires* $O\tilde{~}(r(\nu + \log \max_i \|f_i\|)^2 \cdot \log(1/\epsilon))$ *additional bit operations (using standard arithmetic) and* $O\tilde{~}(s + r(\nu + \log \max_i \|f_i\|))$ *bits of additional storage.*

Application to finding determinantal divisors

Again, let $A \in \mathbb{Z}^{m\times n}$ have rank r. Using `FindContents` with a "black-box" which returns the high-order r coefficients $a_{n-1}, \ldots, a_{n-r} \in \mathbb{Z}[x_2, \ldots, x_n, y_2, \ldots, y_n, z_1, \ldots, z_n]$ of the characteristic polynomial of $\mathcal{U}\hat{A}\mathcal{L}\mathcal{D}$, evaluated at any point in R^{3n-2}, we can identify the contents of $a_{n-1}, \ldots, a_{n-r}$ with provable accuracy. To evaluate $(a_{n-1}(x_i, y_i, z_i), \ldots, a_{n-r}(x_i, y_i, z_i))$ we use the algorithm of Lemma 1.3 at points in $\mathsf{R} \bmod p$. If p does not divide the discriminant of Γ, then $\mathsf{R} \bmod p$ decomposes as a direct sum of finite fields. We apply Lemma 1.3 in each of these fields and use the Chinese remainder algorithm to reconstruct the result in $\mathsf{R} \bmod p$. When this is done for sufficiently many primes p, we can reconstruct the evaluation over R. It is straightforward to show that there are sufficiently many "good" primes p which do not divide the disciminant of Γ and modulo which A does not change rank.

Theorem 2.5. *Let* $A \in \mathbb{Z}^{m\times n}$ *and* $\epsilon > 0$. *The algorithm* `FindContents`, *applied as described above to computing the determinantal divisors* $d_1, \ldots, d_r$ *of* A, *requires* $O\tilde{~}(m^2 \log \|A\| \cdot \log(1/\epsilon))$ *modular matrix-vector products* $Av \bmod (h, p)$, *where* $h \in \mathbb{Z}_p[x]$ *is irreducible of degree* $O(\log m + \log\log \|A\|)$, p *has* $O(\log m + \log\log \|A\|)$ *bits, and* $v \in (\mathbb{Z}_p[z]/(h))^{n\times 1}$. *As well, an additional* $O\tilde{~}((m^2 n \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ *bit operations and storage for* $O\tilde{~}(m^2 \log \|A\| + n)$ *bits are required.*

References

J. Buchmann. A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. In *Séminaire de théorie des nombres*, Paris, 1988.

T. J. Chou and G. E. Collins. Algorithms for the solution of systems of linear Diophantine equations. *SIAM J. of Computing* **11**, pp. 687–708, 1982.

P. Domich, R. Kannan, and L. Trotter. Hermite normal form computation using modulo determinant arithmetic. *Math. Operations Research* **12**, pp. 50–59, 1987.

M. Giesbrecht. Fast computation of the smith form of an integer matrix. In *Proceedings of ISSAC'95*, pp. 110–118, Montreal, Quebec, 1995. ACM Press.

J. L. Hafner and K. S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Amer. Math. Soc.* **2**, pp. 837–850, 1989.

J. L. Hafner and K. S. McCurley. Asymptotically fast triangulization of matrices over rings. *SIAM J. of Computing* **20**(6), pp. 1068–1083, 1991.

G. Havas, D. Holt, and S. Rees. Recognizing badly presented Z-modules. *Linear algebra and its applications* **192**, pp. 137–163, 1993.

C. Iliopolous. 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 J. Computing* **18**, pp. 658–669, 1989.

E. Kaltofen and B. D. Saunders. On Wiedemann's method of solving sparse linear systems. In *Proc. AAECC-9*, vol. 539 of *Springer Lecture Notes in Comp. Sci.*, 1991. 29-38.

R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comp.* **8**, pp. 499–507, 1979.

M. Newman. *Integral Matrices.* Academic Press (New York), 1972.

J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Computing Machinery* **27**, pp. 701–717, 1980.

H. J. S. Smith. On systems of linear indeterminate equations and congruences. *Philos. Trans. Royal Soc. London* **151**, pp. 293–326, 1861.

A. Storjohann, 1995. Personal Communication.

D. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory* **IT-32**, pp. 54–62, 1986.