FAST COMPUTATION OF THE SMITH FORM OF A SPARSE INTEGER MATRIX

MARK GIESBRECHT

Abstract. We present a new probabilistic algorithm 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 $O(m^2 \log \|A\|)$ black box evaluations $w \mapsto Aw \bmod p$ for word-sized primes p and $w \in \mathbb{Z}_p^{n\times 1}$, plus $O(m^2 n \log \|A\| + m^3 \log^2 \|A\|)$ additional bit operations. For sparse matrices this represents a substantial improvement over previously known algorithms. The new algorithm suffers from no "fill-in" or intermediate value explosion, and uses very little additional space. We also present an asymptotically fast algorithm for dense matrices which requires about $O(n \cdot \mathrm{MM}(m) \log \|A\| + m^3 \log^2 \|A\|)$ bit operations, where $O(\mathrm{MM}(m))$ operations are sufficient to multiply two $m \times m$ matrices over a field. Both algorithms are probabilistic of the Monte Carlo type – on any input they return the correct answer with a controllable, exponentially small probability of error.

Key words. Sparse integer matrix, Smith form, probabilistic algorithms
Subject classifications. 15-04, 15A21, 15A36, 11D04

Introduction

It was shown by Smith (1861) that any $A \in \mathbb{Z}^{m\times n}$ is equivalent to a unique diagonal matrix $S \in \mathbb{Z}^{m\times n}$ 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

$$S = PAQ = \begin{pmatrix} s_1 & & & & \\ & \ddots & & 0 & \\ & & s_r & & \\ & & & 0 & \\ & 0 & & & 0 \end{pmatrix} = \mathrm{diag}(s_1, \ldots, s_r, 0, \ldots, 0)$$

Submitted to Computational Complexity: October 20, 1996

where $r = \text{rank}(A)$ and $s_i \mid s_{i+1}$ for $1 \leq i \leq r-1$. S is called the *Smith normal form* of A and $s_1, \ldots, s_r \in \mathbb{Z}\backslash\{0\}$ 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, Chou & Collins 1982), combinatorics (see Wallis *et al.* 1972), and determining the canonical structure of Abelian groups (see Newman 1972). 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).

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 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 in our cost analyses: for any $f, g : \mathbb{R}^l \to \mathbb{R}$, $f = \tilde{O}(g)$ 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 $\|v\| = \|v\|_\infty$, for $B \in \mathbb{Z}^{m\times n}$ we write $\|B\| = \|B\|_\Delta = \max_{ij} |B_{ij}|$, and for $g = \sum_{0\leq i\leq t} b_i x^i \in \mathbb{Z}[x]$ we write $\|g\| = \max_i |b_i|$.

We give two algorithms here. The first is for sparse matrices A, and is based on a combination of techniques developed in Wiedemann (1986), Kaltofen *et al.* (1987, 1990) and Kaltofen & Saunders (1991). An extended abstract of a similar algorithm to this first appeared in Giesbrecht (1996). The second algorithm is for dense matrices and is based on some of the same techniques with asymptotically fast matrix arithmetic replacing Wiedemann's sparse matrix methods. An extended abstract describing an algorithm similar to this appeared in Giesbrecht (1995). In the current paper we present a more uniform and complete exposition of the techniques developed.

Throughout we suppose $A \in \mathbb{Z}^{n\times n}$ has rank r and bottom $n - m$ rows zero, so $r \leq m \leq n$ (so rectangular matrices are embedded in square matrices by padding with rows of zeros to allow for a more uniform treatment; the relative sizes of m and n are exploited by the algorithm and reflected in the analyses). 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 $S = \text{diag}(d_1, d_2/d_1, \ldots, d_r/d_{r-1}, 0, \ldots, 0) \in \mathbb{Z}^{n\times n}$. We use this formulation in

the algorithms here, and compute the determinantal divisors from which the Smith form is trivially constructed.

The first key new idea, presented in Section 1, rests on a new characterization of the determinantal divisors of an integer matrix A. From A we construct a perturbation $\mathfrak{B}$ by pre-multiplying it by a "scaled" Toeplitz matrix of indeterminates (see (1.1), (1.2)). The characteristic polynomial $f = \sum_{0 \le i \le n} f_{n-i} x^i$ of $\mathfrak{B}$ has coefficients $f_0, \ldots, f_r$ which are themselves multi-variate integer polynomials. In Theorem 1.2 we show that the content (in $\mathbb{Z}$) of f_k is exactly the kth determinantal divisor of A. While an explicit representation of f_k is impractical to use computationally (it has exponentially many terms), we can evaluate f quickly at most instantiations to integers of the indeterminates in $\mathfrak{B}$. Thus we think of the coefficients of f as a "black box" which we can evaluate but cannot write down.

The second key idea, presented in Section 2, is to show how to find the contents a list of multi-variate polynomials $(f_1, \ldots, f_r)$ given by a black box. The basic algorithm converges on the contents as a sequence of a logarithmic number l of integer r-tuples $\bar{d}^{(1)}, \ldots, \bar{d}^{(l)} \in \mathbb{Z}^r$. The jth such r-tuple $\bar{d}^{(j)} = (d_1^{(j)}, \ldots, d_r^{(j)}) \in \mathbb{Z}^r$ is constructed from an evaluation of $f_1, \ldots, f_r$ at a random point $\bar{a}$: suppose $(b_1^{(j)}, \ldots, b_r^{(j)}) = (f_1(\bar{a}^{(j)}), \ldots, f_r(\bar{a}^{(j)})) \in \mathbb{Z}^r$. We then set $d_i^{(j)} = \gcd(d_i^{(j-1)}, b_i^{(j)}) \in \mathbb{Z}$. Convergence is p-adic: for a prime p and $1 \le i \le r$,

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

where equality hopefully holds with high probability after the lth iteration. Unfortunately, this algorithm cannot be proven to converge in general. Instead, we choose points randomly in a so-called "rough" extension ring of $\mathsf{R} \supseteq \mathbb{Z}$, a direct sum of orders of number fields. R is specially constructed such that certain primes (those dividing the contents) have relatively high degree in one of the component orders. The degree of $\mathsf{R}/\mathbb{Z}$ can be kept surprisingly small and hence inexpensive to work in. Moreover, the above algorithm with random choices from R can be proven to work efficiently.

In Section 3 we show how to determine efficiently the characteristic polynomial of a random Toeplitz perturbation $B \in \mathsf{R}^{n \times n}$ of A. This is done for sparse matrices using the linear recurrence methods of Wiedemann (1986) and Kaltofen & Saunders (1991) modulo sufficiently many randomly chosen primes. An algorithm is also presented for dense matrices along the lines of the method of Keller-Gehrig (1985). Care is taken to avoid "bad" primes, modulo which the problem changes locally.

Finally, in Section 4 we tie the techniques together into an algorithm for the Smith form. The cost of the sparse algorithm will be measured in terms of

the number of modular matrix-vector products by A required and the number of auxiliary bit operations and space needed. The algorithm requires $\tilde{O}(m^2 \log \|A\| \cdot \log(1/\epsilon))$ modular matrix-vector products $Av \bmod p$ where $v \in \mathbb{Z}_p^{n\times 1}$ and p is a (single-word) prime with $O(\log n + \log\log \|A\|)$ bits. It will also require an additional $\tilde{O}((m^2 n \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ bit operations and storage for $\tilde{O}(m^2 \log \|A\| + n)$ 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 at least $1-\epsilon$. For dense matrices we present an algorithm which requires $\tilde{O}((n\,\mathrm{MM}(m) \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ bit operations and additional storage for $\tilde{O}(nm + m^2 \log \|A\|)$ bits.

For comparison, consider computing the Smith form of an $A \in \mathbb{Z}^{n\times n}$ with $O(n^{1+\xi})$ non-zero entries (for some $\xi : 0 < \xi \leq 1$). The currently best-known (deterministic) algorithm of Storjohann (1996) requires $\tilde{O}(\mathrm{MM}(n) \cdot \mathrm{M}(n \log \|A\|))$ bit operations, where $O(\mathrm{M}(l))$ bit operations are required to multiply two l-bit integers: $\mathrm{M}(l) = l^2$ for standard integer arithmetic. The algorithm also requires $O(nm^2 \log \|A\|)$ bits of additional memory, and takes little advantage of the sparsity of A (the difficult to analyse algorithms of Havas *et al.* (1993), which work well in practice, optimistically have about the same asymptotic complexity as Storjohann's algorithm. Previous algorithms which have been rigorously analysed require at least an additional factor of n in their costs). By contrast, our new algorithm for sparse matrices proposed here requires $\tilde{O}(n^{3+\xi} \log \|A\| + n^3 \log^2 \|A\|)$ bit operations and only $\tilde{O}(n^2 \log \|A\|)$ bits of additional memory. The difference is even more pronounced when m is much smaller than n. Unfortunately, we are not able to compute the transition matrices to the Smith form within this cost. Because these are generally dense with fairly large entries, one cannot expect to compute them in the order of time and space our algorithms require.

Definitions and Notation. For integers n and $k \leq n$, define

$$\mathcal{C}_k^n = \{(c_1, \ldots, c_k) \in \mathbb{N}^k : 1 \leq c_1 < \cdots < c_k \leq n\}.$$

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* $B\left[\begin{smallmatrix}\sigma\\ \tau\end{smallmatrix}\right]$:

$$B\begin{bmatrix}\sigma\\ \tau\end{bmatrix} = \begin{pmatrix} B_{b_1 c_1} & \cdots & B_{b_1 c_k} \\ \vdots & & \vdots \\ B_{b_k c_1} & \cdots & B_{b_k c_k} \end{pmatrix} \in \mathsf{P}^{k\times k},$$

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 characterisation of determinantal divisors

Our new algorithms are based on a new characterization of the determinantal divisors of a matrix. Let $\Lambda = \{v_1, \ldots, v_n, w_1, \ldots, w_n, y_1, \ldots, y_{2n-1}\}$ be a set of algebraically independent indeterminates. Let

$$\mathfrak{D}_1 = \begin{pmatrix} v_1 & & & 0 \\ & v_2 & & \\ & & \ddots & \\ 0 & & & v_n \end{pmatrix}, \quad \mathfrak{D}_2 = \begin{pmatrix} w_1 & & & 0 \\ & w_2 & & \\ & & \ddots & \\ 0 & & & w_n \end{pmatrix} \tag{1.1}$$

and $\mathfrak{T}$ a generic Toeplitz matrix:

$$\mathfrak{T} = \begin{pmatrix} y_n & y_{n+1} & \cdots & y_1 \\ \vdots & y_n & \ddots & \vdots \\ y_{2n-2} & & \ddots & y_{n+1} \\ y_{2n-1} & y_{2n-2} & \cdots & y_n \end{pmatrix}. \tag{1.2}$$

The following lemma describes a useful property of the minors of Toeplitz matrices.

LEMMA 1.1. *For a generic Toeplitz matrix as in* (1.2) *and any* $\sigma, \tau \in \mathcal{C}_k^n$ *(where* $k \leq n$*).* $\mathfrak{T}\binom{\sigma}{\tau} \in \mathbb{Z}[y_1, \ldots, y_{2n-1}]$ *is non-zero with content* 1.

PROOF. For fixed n we prove this by induction on k. If $k = 1$ then $\sigma = (\sigma_1)$, $\tau = (\tau_1)$ and $\mathfrak{T}\binom{\sigma}{\tau} = y_{n+\sigma_1-\tau_1}$, which is clearly non-zero of content 1.

Assume the theorem holds for all $\sigma', \tau' \in \mathcal{C}_{k-1}^n$. Then

$$\begin{aligned} \mathfrak{T}\begin{pmatrix} \sigma \\ \tau \end{pmatrix} &= \sum_{1 \leq i \leq k} (-1)^{i+1} \cdot y_{n+\sigma_i-\tau_1} \cdot \mathfrak{T}\begin{pmatrix} \sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_k \\ \tau_2, \ldots, \tau_k \end{pmatrix} \\ &= (-1)^{k+1} \cdot y_{n+\sigma_k-\tau_1} \cdot \mathfrak{T}\begin{pmatrix} \sigma_1, \ldots, \sigma_{k-1} \\ \tau_2, \ldots, \tau_k \end{pmatrix} \\ &\quad + \sum_{1 \leq i \leq k-1} (-1)^{i+1} \cdot y_{n+\sigma_i-\tau_1} \cdot \mathfrak{T}\begin{pmatrix} \sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_k \\ \tau_2, \ldots, \tau_k \end{pmatrix} \end{aligned}$$

Now $y_{n+\sigma_k-\tau_1} \cdot \mathfrak{T}\binom{\sigma_1, \ldots, \sigma_{k-1}}{\sigma_2, \ldots, \sigma_k}$ is non-zero with content 1. Moreover, $y_{n+\sigma_k-\tau_1}$ does not occur in the remaining terms of the summation. Thus $\mathfrak{T}\binom{\sigma}{\tau}$ is non-zero with content 1. □

THEOREM 1.2. *Let $A \in \mathbb{Z}^{n\times n}$ have rank r and $\mathfrak{B} = \mathfrak{D}_1\mathfrak{T}\mathfrak{D}_2 A$ as above. Let $f = \text{charpoly}(\mathfrak{B}) = \sum_{0\le i\le n} f_{n-i}x^i$ where $f_{n-i} \in \mathbb{Z}[\Lambda]$. Also let $g = \text{minpoly}(\mathfrak{B}) \in \mathbb{Z}[\Lambda][x]$. Then*

(i) $d_k = \text{cont}(f_k)$ for $1 \le k \le n$;

(ii) if $r = n$ then $f = g$ while if $r < n$ then $f = x^{n-r+1}\cdot g$ and g is squarefree with $g(0) = 0$.

PROOF. To prove (i) recall that

$$\begin{aligned} f_k =& (-1)^k \sum_{\sigma\in\mathcal{C}_k^n} \mathfrak{B}\binom{\sigma}{\sigma} = \sum_{\sigma,\nu,\mu,\tau\in\mathcal{C}_k^n} \mathfrak{D}_1\binom{\sigma}{\nu}\mathfrak{T}\binom{\nu}{\mu}\mathfrak{D}_2\binom{\mu}{\tau}A\binom{\tau}{\sigma} \\ =& \sum_{\sigma,\tau\in\mathcal{C}_k^n} \mathfrak{D}_1\binom{\sigma}{\sigma}\mathfrak{T}\binom{\sigma}{\tau}\mathfrak{D}_2\binom{\tau}{\tau}A\binom{\tau}{\sigma} \\ =& \sum_{\sigma,\tau\in\mathcal{C}_k^n} v_{\sigma_1}\cdots v_{\sigma_k}w_{\tau_1}\cdots v_{\tau_k}\mathfrak{T}\binom{\sigma}{\tau}A\binom{\tau}{\sigma} \end{aligned}$$

using the Binet-Cauchy formula. By Lemma 1.1, $\mathfrak{T}\binom{\sigma}{\tau}$ is non-zero with content 1. Thus, for each pair σ,τ there is a term in f_k with coefficient $A\binom{\sigma}{\tau}$ and the coefficient of every term is a $k\times k$ minor of A. Therefore

$$\text{cont}(f_k) = \gcd\left\{A\binom{\sigma}{\tau} : \sigma,\tau\in\mathcal{C}_k^n\right\} = d_k.$$

To prove (ii), first recall that $\text{cont}(f_l) = d_l = 0$ for $l > r$, whence $f = x^{n-r}\bar{g}$ with $\bar{g}(0) \ne 0$. If $r = n$ then $f = g = \bar{g}$. If $r < n$ and $\text{disc}(\bar{g}) \ne 0$ then $g = x\cdot\bar{g}$ and $\text{disc}(\bar{g}) = \text{disc}(g)$. In either case it is sufficient to show that $\text{disc}(\bar{g}) \ne 0$ Note that

$$\text{charpoly}(\mathfrak{D}_1\mathfrak{T}\mathfrak{D}_2 A) = \text{charpoly}(\mathfrak{T}\mathfrak{D}_2 A\mathfrak{D}_1).$$

Let $\sigma = (\sigma_1,\ldots,\sigma_r)$, $\tau = (\tau_1,\ldots,\tau_r) \in \mathcal{C}_r^n$ be such that $A\binom{\sigma}{\tau} \ne 0$. We next show that $(\mathfrak{T}\mathfrak{D}_2 A)\binom{\tau_1,\ldots,\tau_k}{\tau_1,\ldots,\tau_k} \ne 0$ for $1 \le k \le r$.

$$(\mathfrak{T}\mathfrak{D}_2 A)\binom{\tau_1,\ldots,\tau_k}{\tau_1,\ldots,\tau_k} = \sum_{\mu\in\mathcal{C}_k^n} w_{\mu_1}w_{\mu_2}\cdots w_{\mu_k}\cdot\mathfrak{T}\binom{\tau_1,\ldots,\tau_k}{\mu_1,\ldots,\mu_k}A\binom{\mu_1,\ldots,\mu_k}{\tau_1,\ldots,\tau_k}.$$

By Lemma 1.1 all minors of $\mathfrak{T}$ are non-zero of content 1. Moreover, since $A\binom{\sigma}{\tau} \ne 0$ (and hence columns $\tau_1,\ldots,\tau_k$ of $A\left[\begin{smallmatrix}\sigma\\ \tau\end{smallmatrix}\right]$ are linearly independent),

there exists a $\mu \in \mathcal{C}_k^n$ such that $A\binom{\mu_1,\ldots,\mu_k}{\tau_1,\ldots,\tau_k} \neq 0$. Hence $(\mathfrak{T}\mathfrak{D}_2 A)\binom{\tau_1,\ldots,\tau_k}{\tau_1,\ldots,\tau_k} \neq 0$ for $1 \leq k \leq r$.

To show that discriminant of $\bar{g}$ is non-zero of the desired degree, set $v_i := 0$ for $i \in \{1,\ldots,n\} \setminus \{\tau_1,\ldots,\tau_k\}$ to form $\hat{\mathfrak{D}}_1^{(k)}$ from $\mathfrak{D}_1$. Then

$$\text{charpoly}(\mathfrak{T}\mathfrak{D}_2 A\hat{\mathfrak{D}}_1^{(k)}) = x^{n-r} \cdot \text{charpoly}\left((\mathfrak{T}\mathfrak{D}_2 A\hat{\mathfrak{D}}_1^{(k)})\begin{bmatrix}\tau\\ \tau\end{bmatrix}\right).$$

Since $(\mathfrak{T}\mathfrak{D}_2 A)\left[\begin{smallmatrix}\tau\\ \tau\end{smallmatrix}\right]$ has all leading minors non-zero, $h = \text{charpoly}(\mathfrak{T}\mathfrak{D}_2 A\hat{\mathfrak{D}}_1^{(k)})\left[\begin{smallmatrix}\tau\\ \tau\end{smallmatrix}\right]$ has a non-zero discriminant by Wiedemann (1986). Since the discriminant of h is simply the discriminant of $\bar{g}$ with some of the v_i's set to 0 as above, $\text{disc}(\bar{g}) \neq 0$. □

2. Finding the content of black-box polynomials

In this section we give an algorithm 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, to find the determinantal divisors of an integer matrix it is sufficient to be able to find the contents of a sequence of integer polynomials

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

In the terminology of Section 1, f_i is the coefficient of x^{n-i} of the characteristic polynomial of $\mathfrak{B}$, with content d_i, the ith determinantal divisor of A, $s = 4n-1$ and $\deg f_i \leq \nu$ where $\nu = 3m$.

We assume 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.

Informally, the idea is as follows. We maintain a vector $(c_1,\ldots,c_r) \in \mathbb{Z}^r$ which contains an "approximation" to $(\text{cont}(f_1),\ldots,\text{cont}(f_r))$. 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 $d_i = \text{cont}(f_i) \in \mathbb{Z}$ 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 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 — the "Schwartz-Zippel Lemma" (Schwartz 1980, Zippel 1979) — 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 polynomial $\Gamma \in \mathbb{Z}[z]$ of logarithmic degree and set $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$. We construct Γ such that $\Gamma = \Gamma_1 \Gamma_2 \cdots \Gamma_k$ where $\Gamma_i \in \mathbb{Z}[z]$ are distinct, 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 an order in the number field $\mathbb{Q}[z]/(\Gamma_i)$. For each small p dividing c_i we ensure there exists a j such that $\mathsf{O}_j \bmod p$ contains a sufficiently large finite field. Equivalently, $\Gamma_j \bmod p$ has a sufficiently large irreducible factor in $\mathbb{Z}_p[x]$. This ensures that the probability of $g_i(\vec{a}) \not\equiv 0 \bmod p$ is high for $\vec{a}$ chosen "randomly" from R^s. See Lemmas 2.6 and 2.7 below. That such an extension can be built efficiently is shown in Theorem 2.3 below.

We perform evaluations at points in $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$ instead of $\mathbb{Z}$. While GCD's are not necessarily well defined in R we instead take the GCD's of the *contents* of the evaluations, treating them as polynomials in $\mathbb{Z}[z]$ of degree less than $\deg \Gamma$. For $a \in \mathsf{R}$ we define the content $\overline{\mathrm{cont}}(a) = \mathrm{cont}(\bar{a}) \in \mathbb{Z}$, where $\bar{a} \in \mathbb{Z}[z]$, $\bar{a} \equiv a \bmod \Gamma$ and $\deg \bar{a} < \deg \Gamma$. Equivalently, $\overline{\mathrm{cont}}(a)$ is the largest rational integer c such that $a \equiv 0 \bmod c$.

Consider, for example, finding the content of $f(x) = 5x^4 + 10x^3 - 5x^2 - 10x$: clearly $\mathrm{cont}(f) = 5$. It is easily shown that $f(a) \equiv 0 \bmod 120$ for all $a \in \mathbb{Z}$,

so we cannot identify the content by computing GCD's of evaluations of f at integers. We instead perform our evaluations in $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$ where $\Gamma = z^3+2z^2+2z+3 \in \mathbb{Z}[z]$. Randomly choose $a_1 = (z^2+z \bmod \Gamma) \in \mathsf{R}$ and compute $f(a_1) \equiv 20z^2 + 20z \bmod \Gamma$. Then $\overline{\text{cont}}(f(a_1)) = 20$. Now choose $a_2 = z^2 + 2z$ and obtain $f(a_2) \equiv 105z^2 + 255z + 120 \bmod \Gamma$ and $\overline{\text{cont}}(f(a_2)) = 15$. Finally $\gcd(\overline{\text{cont}}(f(a_1)), \overline{\text{cont}}(f(a_2))) = 5$, the correct content of f.

A non-uniform variant of the Schwartz-Zippel Lemma. To prove that the algorithm sketched above to find the contents of black-box polynomials converges we will require a variant of the Schwartz-Zippel Lemma. This bounds from above the probability that a random point, with coordinates chosen randomly and uniformly from a finite set $\mathcal{V}$, is a non-zero of a polynomial. We require a version of this in which coordinates are not chosen uniformly from $\mathcal{V}$, but where we only have an upper bound on the probability of choosing any one element of $\mathcal{V}$.

LEMMA 2.1. *Assume $f \in \mathsf{D}[x_1, \ldots, x_k]$ is non-zero, D an integral domain, and $\mathcal{V}$ a finite subset of D. Suppose elements $a_1, \ldots, a_k$ are randomly chosen from $\mathcal{V}$ such that each a_i is assigned any one element of $\mathcal{V}$ with probability at most ϱ. Then* $\text{Prob}\{f(a_1, \ldots, a_k) = 0 : a_1, \ldots, a_k \in \mathcal{V}\} \leq \varrho \deg f$.

PROOF. We proceed by induction on k. For $k = 1$, f has at most $\deg f$ roots, hence $\text{Prob}\{f(a_1) = 0 : a_1 \in \mathcal{V}\} \leq \varrho \deg f$. Assume the inductive hypothesis that the lemma holds for $k-1$. We now show it for k. Write

$$f(x_1, \ldots, x_k) = g(x_1, \ldots, x_{k-1})x_k^d + h(x_1, \ldots, x_k),$$

where $g \in \mathsf{D}[x_1, \ldots, x_{k-1}]$ and $h \in \mathsf{D}[x_1, \ldots, x_k]$, with $\deg_{x_k} h < d$ (i.e., h consists of the non-leading terms of f considered as a polynomial in x_k). Then $\deg g \leq \deg(f) - d$ and by induction

$$\text{Prob}\{g(a_1, \ldots, a_{k-1}) = 0 : a_1, \ldots, a_{k-1} \in \mathcal{V}\} \leq \varrho(\deg(f) - d).$$

In the case $g(a_1, \ldots, a_{k-1}) \neq 0$, there are at most d roots for $f(a_1, \ldots, a_{k-1}, x_k)$. Therefore

$$\begin{aligned}
&\text{Prob}\{f(a_1 \ldots, a_k) = 0 : a_1, \ldots, a_k \in \mathcal{V}\} \\
&\quad = \text{Prob}\{f(a_1, \ldots, a_k) = 0 \mid g(a_1, \ldots, a_{k-1}) = 0\} \cdot \text{Prob}\{g(a_1, \ldots, a_{k-1}) = 0\} \\
&\qquad + \text{Prob}\{f(a_1, \ldots, a_k) = 0 \mid g(a_1, \ldots, a_{k-1}) \neq 0\} \cdot \text{Prob}\{g(a_1, \ldots, a_{k-1}) \neq 0\} \\
&\quad \leq \text{Prob}\{g(a_1, \ldots, a_{k-1}) = 0\} + \text{Prob}\{f(a_1, \ldots, a_k) = 0 \mid g(a_1, \ldots, a_{k-1}) \neq 0\} \\
&\quad \leq \varrho(\deg(f) - d) + \varrho d = \varrho \deg f.
\end{aligned}$$

□

Building a rough extension of $\mathbb{Z}$. Unfortunately Lemma 2.1 above, like the Schwartz-Zippel Lemma, is of little use when the size of $\mathcal{V}$ is less than $\deg f$. In particular, when $\mathsf{D} = \mathbb{Z}_p$, $\#\mathcal{V} \leq p$ and $\tau \geq 1/p$. When $\deg f \geq p$, Lemma 2.1 is trivial. Our solution will be to construct a *rough* extension ring R of $\mathbb{Z}$, one such that $\mathsf{R} \bmod p$ contains a large finite field for each of selected set of primes p. We show that the degree of R over $\mathbb{Z}$ can be kept surprisingly small.

Let $\eta \geq 1$ and $p_1, \ldots, p_\kappa$ be primes with $2 \leq p_1, \ldots, p_\kappa \leq \tau$ for some $\tau \geq 2$. We next show how to construct a monic, squarefree $\Gamma \in \mathbb{Z}[x]$ of small height and degree $\varrho = O(\eta \log \kappa)$ such that for each i $(1 \leq i \leq \kappa)$, $\Gamma \bmod p_i$ has a factor of degree greater than η in $\mathbb{Z}_p[x]$ (i.e., the factorization of Γ is "rough" modulo each prime p_i). Notice that the degree of Γ is *logarithmic* in the number of primes.

For any fixed prime p and $n \in \mathbb{N}$, define

$$\mathcal{M}_p(n) = \{g \in \mathbb{Z}_p[x] : g \text{ monic, squarefree, } \deg g = n\},$$
$$\mathcal{I}_p(n) = \{g \in \mathbb{Z}_p[x] : g \text{ monic, irreducible, } \deg g = n\}.$$

We first give a lower bound on

$$R_p(\eta) = \#\{f \in \mathcal{M}_p(2\eta) : \exists k > \eta,\ g \in \mathcal{I}_p(k) \text{ such that } g \mid f\},$$

the number of monic, squarefree polynomials in $\mathbb{Z}_p[x]$ of degree 2η which have an irreducible factor in $\mathbb{Z}_p[x]$ of degree greater than η.

LEMMA 2.2. *For a prime p and integer $\eta \geq 1$ we have $R_p(\eta) > p^{2\eta}/5$.*

PROOF. Any $f \in \mathcal{M}_p(2\eta)$ can have at most one irreducible factor with degree greater than η. Also, the number of squarefree polynomials of degree j in $\mathbb{Z}_p[x]$ is $p^j(1 - 1/p)$ for any j. We obtain the formula

$$R_p(\eta) = \sum_{\eta < i \leq 2\eta} p^{2\eta - i}(1 - 1/p) \cdot N_p(i)$$

where $N_p(i) = \#\mathcal{I}_p(i)$. Using the lower bound on $N_p(i)$ given by Lidl & Niederreiter(1983, Exercise 3.27) and Euler's summation formula, we get

$$R_p(\eta) > p^{2\eta} \cdot (1 - 1/p) \cdot \left(\sum_{\eta < i \leq 2\eta} \frac{1}{i} - \frac{p}{p-1} \sum_{\eta < i \leq 2\eta} \frac{1}{ip^{i/2}} \right) > p^{2\eta} \cdot s(p, \eta),$$

where

$$s(p, \eta) = (1 - 1/p) \cdot \left(\log 2 - \frac{1}{2\eta} - \frac{p}{p-1} \cdot \frac{1}{\eta} \cdot \frac{\sqrt{p}}{\sqrt{p} - 1} \cdot \frac{1}{p^{\eta/2 + 1/2}} \right).$$

The function $s(p,\eta)$ is strictly increasing in both p and η and is greater than $1/5$ with the exceptions $p = 2$, $2 \le \eta \le 4$, and $p = 3$, $\eta = 2$. Excepting these cases, $R_p(\eta) > p^{2\eta}/5$. It is easily checked, using the exact formula $N_p(i) = (1/i) \cdot \sum_{d \mid i} \mu(d) p^{i/d}$ for $i \ge 1$ (where μ is the Mobius function), that indeed $R_p(\eta) > p^{\eta}/5$ in the exceptional cases as well. □

Let $\mathcal{V} := \{g \in \mathbb{Z}[x] : g \text{ monic, } \deg g = 2\eta, \text{ and } \|g\| \le 2\eta\tau\}$. If we choose f randomly and uniformly from $\mathcal{V}$, f falls into a particular residue class in $\mathcal{M}_p(2\eta)$ with probability at least $(\lfloor (4\eta\tau+1)/p \rfloor / (4\eta\tau+1))^{2\eta} \ge (1/p - 1/(4\eta\tau+1))^{2\eta}$. Thus by Lemma 2.2 the probability that $f \bmod p$ has an irreducible factor modulo p of degree greater than η is at least

$$\left(\frac{1}{p} - \frac{1}{4\eta\tau+1}\right)^{2\eta} \cdot \frac{p^{2\eta}}{5} = \frac{1}{5} \cdot \left(1 - \frac{p}{4\eta\tau+1}\right)^{2\eta} \ge \frac{1}{5} \cdot \left(1 - \frac{\tau}{4\eta\tau+1}\right)^{2\eta}$$
$$> \frac{1}{5} \cdot \left(1 - \frac{1}{4\eta}\right)^{2\eta} > 1/9.$$

The following algorithm constructs a $\Gamma \in \mathbb{Z}[x]$ as required.

Algorithm: `BuildRoughExtension`

Input: $\eta \in \mathbb{Z}$ and primes $2 \le p_1, \ldots, p_\kappa \le \tau$;

Output: a squarefree, monic $\Gamma \in \mathbb{Z}[x]$ such that for each i $(1 \le i \le \kappa)$, $\Gamma \bmod p_i$ has an irreducible factor in $\mathbb{Z}_{p_i}[x]$ of degree greater than η.

(1) Repeat
(2) Let $\mathcal{P} := \{1, \ldots, \kappa\}$; $H := \{\}$;
(3) For $i := 1$ to $l := 6 + 9 \log \kappa$ while $\mathcal{P} \ne \{\}$ do
(4) Choose a random $h_i \in \mathcal{V}$;
(5) For $j \in \mathcal{P}$ do
(6) If $h_i \bmod p_j \in \mathbb{Z}_{p_j}[x]$ has an irreducible factor modulo p_j of degree greater than η Then $\mathcal{P} := \mathcal{P} \setminus \{j\}$; $H := H \cup \{h_i\}$;
End For;
End For;
Until $\mathcal{P} = \{\}$;
(7) Return $\Gamma = \prod_{h \in H} h \in \mathbb{Z}[x]$;

THEOREM 2.3. *The algorithm* `BuildRoughExtension` *always produces the correct results as described and requires an expected number of* $O((\eta^3+\eta^2\log\tau)\cdot\kappa\log^2\tau\log\kappa)$ *bit operations. The output* $\Gamma\in\mathbb{Z}[x]$ *has degree* $2\eta(6+9\log\kappa)=O(\eta\log\kappa)$ *and* $\|\Gamma\|=(\eta\tau)^{O(\log\kappa)}$.

PROOF. We first examine the probability that the algorithm successfully finds a Γ in an iteration of the outer loop, or equivalently, finds an $H\subseteq\mathcal{V}$ such that for each i, there exists an $h\in H$ such that $h \bmod p_i$ has an irreducible factor in $\mathbb{Z}_p[x]$ of degree greater than η. For fixed j, the probability that $h_i \bmod p_j$ has no factor in $\mathcal{I}_{p_j}(r)$ for some $r>\eta$, for *all* $1\le i\le l$ is less than $(8/9)^l$ by Lemma 2.2. The probability this is true for all j is less than $\kappa\cdot(8/9)^l<1/2$ by our choice of $l=6+9\log\kappa$.

For each random choice of $h_i\in\mathcal{V}$ the inner loop of steps (5)–(6) can be accomplished with an expected number of $O((\eta^3+\eta^2\log\tau)\log^2\tau\cdot\kappa)$ bit operations using Berlekamp's (1970) factoring algorithm, and this loop is executed $6+9\log\kappa$ times per iterations of the outer loop.

For any $h_1,h_2\in\mathbb{Z}[x]$, $\|h_1h_2\|\le\min(\deg h_1,\deg h_2)\cdot\|h_1\|\|h_2\|$. Since Γ is the product of $O(\log\kappa)$ polynomials of degree 2η and height at most $2\eta\tau$, it follows that $\deg\Gamma=2\eta(6+9\log\kappa)$ and $\|\Gamma\|=(\eta\tau)^{O(\log\kappa)}$. □

We define $\mathsf{R}=\mathbb{Z}[x]/(\Gamma)$, an extension ring of R where $\Gamma\in\mathbb{Z}[x]$ is monic of degree ϱ and is constructed using `BuildRoughExtension` on some η and primes $p_1,\ldots,p_\kappa$. The ring $\mathsf{R}_{p_i}=\mathsf{R} \bmod p_i$ contains a copy of $\mathrm{GF}(p_i^{\xi_i})$ for some $\xi_i>\eta$ for $1\le i\le\kappa$. We represent an element $a\in\mathsf{R}$ by its *least degree residue* $\hat{a}\in\mathbb{Z}[x]$ with $a\equiv\hat{a} \bmod\Gamma$ and $\deg\hat{a}<\varrho$. The notion of height is extended to R by $\|a\|=\|\hat{a}\|$.

LEMMA 2.4 *(Giesbrecht 1993). Let* R *be as above. Then*

(i) for $a,b\in\mathsf{R}$, $\|ab\|\le\|a\|\cdot\|b\|\cdot\varrho\cdot\|2\Gamma\|^\varrho$;

(ii) for $X\in\mathsf{R}^{l\times m}$ *and* $Y\in\mathsf{R}^{m\times n}$, $\|XY\|\le m\|A\|\cdot\|B\|\cdot\varrho\cdot\|2\Gamma\|^\varrho$.

LEMMA 2.5. *Let* $f\in\mathbb{Z}[x_1,\ldots,x_s]$ *with* $\deg f\le\nu$ *and* $\vec{a}\in\mathsf{R}^s$, *where* R *is as above. Then* $\log\|f(\vec{a})\|=O(\log\|f\|+\nu\log(\nu+s)+\nu\log\|\vec{a}\|+\varrho\nu\log\|\Gamma\|)$

PROOF.

$$\begin{aligned}\|f(\vec{a})\|\le&\|f\|\sum_{0\le j\le\nu}\binom{j+s-1}{j}\|\vec{a}\|^j\varrho^{j-1}\|2\Gamma\|^{\varrho(j-1)}\\=&O(\|f\|(\nu+s)^\nu\|\vec{a}\|^\varrho\varrho^\nu\|2\Gamma\|^{\nu\varrho}),\end{aligned}$$

by Lemma 2.4 The lemma follows by taking the logarithm of both sides. □

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 finite subset $\mathcal{W}$ of a rough extension ring R of $\mathbb{Z}$ at which to evaluate $f_1, \ldots, f_r$. Let $\eta = 4 \log \nu$ and let $\Gamma \in \mathbb{Z}[z]$ be monic and squarefree of degree $\varrho > 1$ and $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$. Let $\lambda \geq \max(6, \nu^2)$ and

$$\mathcal{H} = \{-\lambda, \ldots, \lambda\} \subseteq \mathbb{Z}, \qquad \mathcal{W} = \{h \bmod \Gamma : h \in \mathcal{H}[z], \deg h < \eta\} \subseteq \mathsf{R}.$$

We will further specify our choices of Γ, ϱ, and λ in the sequel.

In the next two lemmas we examine the probability that, for $g \in \mathbb{Z}[x_1, \ldots, x_s]$ with content 1, a prime p, and a randomly and uniformly selected point $\vec{a} \in \mathcal{W}^s$, that $g(\vec{a}) \equiv 0 \bmod p$.

LEMMA 2.6. *Let $p > 2\lambda$ be prime and $g \in \mathbb{Z}[x_1, \ldots, x_s]$ with $\deg g \leq \nu$ and $\mathrm{cont}(g) = 1$. For randomly chosen $\vec{a} \in \mathcal{W}^s$, Prob $\{g(\vec{a}) \equiv 0 \bmod p\} < 1/\nu$.*

PROOF. Assume $\vec{a} = (a_1, \ldots, a_s) \in \mathcal{W}^s$ where $a_i = \sum_{0 \leq j < \eta} a_{ij} z^j \in \mathbb{Z}[z]$ for $a_{ij} \in \mathcal{H}$. Then $g(\vec{a}) \equiv 0 \bmod p \iff g(a_1, \ldots, a_s) \equiv 0 \bmod (\Gamma, p)$. Assume for now that the a_{ij}'s are independent indeterminates over $\mathbb{Q}$. Let $\Lambda = \{a_{ij} : 1 \leq i \leq s, 0 \leq j < \eta\}$, and define $\hat{g} = g(a_1, \ldots, a_s) \in \mathbb{Z}[\Lambda][z]$. Consider the division of $\hat{g}$ by Γ in $\mathbb{Q}(\Lambda)[z]$ to obtain remainder $\varrho = \sum_{0 \leq k < \varrho} \varrho_k(\Lambda) z^k \in \mathbb{Z}[\Lambda][z]$, where $\varrho_k \in \mathbb{Z}[\Lambda]$ has degree at most ν for $0 \leq k < \varrho$. Now $g \not\equiv 0 \bmod p$ and $p > \deg g$, so there exists a $\vec{b} \in \mathbb{Z}^s$ such that $g(\vec{b}) \not\equiv 0 \bmod p$. Since $g(\vec{b}) \in \mathbb{Z}$, $\varrho_0 \not\equiv 0 \bmod p$.

Again assume $\vec{a}$ is randomly selected from $\mathcal{W}^s$. A necessary condition for $g(\vec{a}) \equiv 0 \bmod p$ is that $\varrho_0(\vec{a}) \equiv 0 \bmod p$, whence

$$\text{Prob}\,\{g(\vec{a}) \equiv 0 \bmod p\} \leq \text{Prob}\,\{\varrho_0(\vec{a}) \equiv 0 \bmod p\} \leq \nu/(2\lambda + 1) < 1/\nu$$

by Corollary 1 of Schwartz (1980). □

LEMMA 2.7. *Let $\lambda \geq 6$ and $p \leq 2\lambda$ prime, and assume $\Gamma \bmod p$ has an irreducible factor $\Upsilon \in \mathbb{Z}_p[x]$ of degree greater than η. Let $g \in \mathbb{Z}[x_1, \ldots, x_s]$ with $\deg g \leq \nu$ and $\mathrm{cont}(g) = 1$. For a randomly and uniformly chosen $\vec{a} \in \mathcal{W}^s$, Prob$\{g(\vec{a}) \equiv 0 \bmod p\} \leq 1/\nu$.*

PROOF. A randomly chosen $a \in \mathcal{W}$ lies in a particular residue class of $\mathcal{W}$ mod p with probability at most

$$\left(\left\lceil \frac{2\lambda + 1}{p} \right\rceil \cdot \frac{1}{2\lambda + 1}\right)^{\eta} \leq \left(\frac{1}{p} + \frac{1}{2\lambda + 1}\right)^{\eta} \leq (3/5)^{\eta}$$

for $\lambda \geq 6$. Since $\deg \Upsilon \geq \eta$ and each $a \in \mathcal{W}$ has $\deg a < \eta$, the probability that a is in a particular residue class of $\mathcal{W}$ mod (p, Υ) is also at most $(3/5)^\eta$. Applying Lemma 2.1, $\text{Prob}\{g(\vec{a}) \equiv 0 \bmod (p, \Upsilon)\} \leq \nu \cdot (3/5)^\eta < 1/\nu$ by our choice of $\eta = 4 \log \nu$. □

Define the extension ring $\overline{\mathsf{R}} = \mathbb{Q} \otimes \mathsf{R} = \mathbb{Q}[z]/(\Gamma)$, with units $\overline{\mathsf{R}}^* = \{a \bmod \Gamma : a \in \mathbb{Q}[z], \gcd(a, \Gamma) = 1\}$. In the definition of our black box for evaluating $(f_1(\vec{a}), \ldots, f_r(\vec{a}))$ at a point $\vec{a} \in \mathsf{R}^s$ 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 $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ the black box may report "failure" and is not evaluated. The next lemma shows that the black box seldom fails.

LEMMA 2.8. *Let $\chi \in \mathbb{Z}[x_1, \ldots, x_s] \backslash \{0\}$. For a randomly and uniformly chosen $\vec{a} \in \mathcal{W}^s$, $\text{Prob}\{\chi(\vec{a}) \notin \overline{\mathsf{R}}^*\} \leq \deg(\chi) \cdot \deg(\Gamma)/(2\lambda)$.*

PROOF. Assume $\vec{a} = (a_1, \ldots, a_s) \in \mathcal{W}^s$ where $a_i = \sum_{0 \leq j < \eta} a_{ij} z^j \in \mathbb{Z}[z]$ for $1 \leq i \leq s$. Then $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ if and only if $\gcd(\chi(a_1, \ldots, a_s), \Gamma) \neq 1$ in $\mathbb{Q}[z]$, which is true if and only if the resultant $\text{Res}(\chi(a_1, \ldots, a_s), \Gamma)$ is zero.

Now assume the a_{ij}'s above are algebraically independent indeterminates over $\mathbb{Q}$ and let $\Lambda = \{a_{ij} : 1 \leq i \leq l, 0 \leq j < \eta\}$. Then $a_i = \sum_{0 \leq j < \eta} a_{ij} z^j \in \mathbb{Z}[\Lambda][z]$ and $\text{Res}(\chi(a_1, \ldots, a_s), \Gamma) \in \mathbb{Z}[\Lambda]$ has degree at most $\deg(\chi) \cdot \deg(\Gamma)$. If the a_{ij}'s are assigned random values from $\mathcal{H}$, then by Schwartz's (1980) Corollary 1, the probability that $\text{Res}(\chi(a_1, \ldots, a_s), \Gamma) = 0$, and hence the probability that $\chi(a_1, \ldots, a_s) \notin \overline{\mathsf{R}}^*$, is at most $\deg(\chi) \cdot \deg(\Gamma)/(2\lambda + 1)$. □

Algorithm: `FindContents`

Input:
- $r \geq 2$, $\epsilon > 0$;
- 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 $f_1, \ldots, f_r \in \mathbb{Z}[x_1, \ldots, x_s]$, $\deg f_i \leq \nu$, 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 $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ the black box may report "failure" and is not evaluated. We also allow that on an input $\vec{a}$, any invocation of the black-box may return "failure" with probability at most $1/2$.

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

(1) Let $\lambda := \max(r\nu + \deg \chi, 100\nu \log(\nu) \cdot \deg(\chi) \log(\nu \deg(\chi)) + 100000, \nu^2)$;

(2) Choose a random $\vec{a} \in \mathcal{H} = \{-\lambda, \ldots, \lambda\}^s$;
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 $< \lambda$ which divide $\operatorname{lcm}(c_1^{(0)}, \ldots, c_r^{(0)})$;

(4) Using `BuildRoughExtension` on $p_1, \ldots, p_\kappa$ and $\eta = 4 \log \nu$ construct $\Gamma \in \mathbb{Z}[z]$, such that for $1 \leq i \leq \kappa$, $\Gamma \bmod p_i$ 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 $l := (\log(1/\epsilon) + \log r + \log\log \max_i |c_i^{(0)}|)/\log(2\nu/(\nu+4))$ Do

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

(7) Evaluate $(b_1, \ldots, b_r) := (f_1(\vec{a}), \ldots, f_r(\vec{a})) \in \mathsf{R}^r$ using the black box; If the black box reports a failure on $\vec{a}$, this choice may be ignored, and execution continued with next i at step (6);

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

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

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

PROOF. Step (2) finds a non-zero multiple c_i of d_i for $1 \leq i \leq r$. For a randomly chosen $\vec{a} \in \{-\lambda, \ldots, \lambda\}$

$$\begin{aligned}\operatorname{Prob}\{\chi(\vec{a}) \neq 0, f_i(\vec{a}) \neq 0 \text{ for all } 1 \leq i \leq r\} &= \operatorname{Prob}\{(\chi \cdot f_1 \cdots f_r)(\vec{a}) \neq 0\} \\ &\geq 1 - (r\nu + \deg \chi)/(2\lambda + 1) \geq 1/2\end{aligned}$$

by Corollary 1 of Schwartz (1980). Thus we expect to evaluate the black box two times on points of height $O(\lambda)$ in step (2). On completion of (2) we have

$$\log |c_i^{(0)}| = \log |f_i(\vec{a})| = O(\log \|f_i\| + \nu \log \lambda + \nu \log(\nu + s)),$$

by Lemma 2.5 (setting $\mathsf{R} = \mathbb{Z}$: $\Gamma = z$ and $\varrho = 1$).

Assume for now we choose $\vec{a}$ randomly in step (6) from $\mathcal{W}$ and do not eliminate the cases when $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ in step (7), or when false failures are reported. Fix an i between 1 and r. For a prime p dividing d_i,

$$\text{Prob}\{\text{ord}_p(\bar{b}_i) \neq \text{ord}_p(d_i)\} = \text{Prob}\,\{g_i(\vec{a}) \equiv 0 \bmod p\} \leq 1/\nu$$

by Lemmas 2.6 and 2.7.

The probability that $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ (which will cause the black box not to work on a particular input) is at most

$$\deg(\chi)\deg(\Gamma)/(2\lambda) = 4\deg(\chi)\cdot\log(\nu)\cdot(4+6\log(\lambda))/\lambda \leq 1/\nu,$$

for $\lambda \geq 100\nu\log(\nu)\cdot\deg(\chi)\log(\nu\deg(\chi)) + 1000000$ using Lemma 2.8 and the fact that $\kappa \leq \lambda$. Also, on any input, for any invocation of the black box, it may report failure with probability at most $1/2$. For a fixed i and prime p dividing $c_i^{(0)}$, the probability that $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ *or* $\text{ord}_p(\bar{b}_i) \neq \text{ord}_p(d_i)$ is at most $1/2+2/\nu$.

Let $\omega(c_i^{(0)})$ denote the number of distinct primes dividing $c_i^{(0)}$. The probability that after l iterations of the loop there exists an i and a prime p dividing $c_i^{(0)}$ such that for all l randomly chosen $\vec{a} \in \mathsf{R}^s$ we have $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ or $\text{ord}_p(c_i^{(l)}) \neq \text{ord}_p(d_i)$ is at most

$$\sum_{1\leq i\leq r} \omega(c_i^{(0)})\cdot(1/2+2/\nu)^l \leq r\log_2\max|c_i^{(0)}|(1/2+2/\nu)^l \leq \epsilon$$

by our choice of l. Thus, the probability that the algorithm produces the correct answer is at least $1-\epsilon$, as required. The number l of evaluations of the black box has order $O(\log(1/\epsilon)+\log r+\log\log s+\log\log\max_i\|f_i\|)$.

Note that $\varrho = O(\eta\log\kappa) = O(\log(\nu)\log(r))$ and $\log\|\Gamma\| = O((\log r + \log\nu)^2)$ by Lemma 2.3 (recall $\deg\chi = O(\nu^2)$). Arguments $(a_1,\ldots,a_s) \in \mathcal{W}^s$ to the black box satisfy $\|a_i\| \leq \lambda = (r\nu)^{O(1)}$ for $1 \leq i \leq s$.

The cost (in addition to the cost of the black-box evaluations) is dominated by the loop in steps (5)-(9). For $1 \leq i \leq r$,

$$\log\|b_i\| = O(\log\|f_i\| + \nu\log(\nu+s) + \nu\log(\nu)\log(r)\cdot(\log(r)+\log(\nu))^2)$$

by Lemma 2.5. The cost of l iterations of (8)-(9) is $\tilde{O}(lr(\nu+\log\|f_i\|)^2)$ bit operations using standard arithmetic. An additional $\tilde{O}(ls)$ bit operations are required by step (6). At any time we store $O(r)$ elements of R with $\tilde{O}(\nu+\log\max_i\|f_i\|)$ bits each and s elements with $O(\log r+\log\nu)$ bits. □

3. Computing the characteristic polynomial

In this section we describe an algorithm to compute the characteristic polynomial of a matrix $B \in \mathsf{R}^{n\times n}$. As in the previous section, $\mathsf{R} = \mathbb{Z}[z]/(\Gamma)$ where $\Gamma \in \mathbb{Z}[z]$ is monic and squarefree of degree ϱ. The ring R decomposes as

$$\mathsf{R} = \frac{\mathbb{Z}[z]}{(\Gamma_1)} \oplus \frac{\mathbb{Z}[z]}{(\Gamma_2)} \oplus \cdots \oplus \frac{\mathbb{Z}[z]}{(\Gamma_k)} \subseteq \mathsf{E}_1 \oplus \mathsf{E}_2 \oplus \cdots \oplus \mathsf{E}_k,$$

where $\Gamma = \Gamma_1 \cdots \Gamma_k$, and $\Gamma_1, \ldots, \Gamma_k \in \mathbb{Z}[z]$ are distinct, monic and irreducible in $\mathbb{Z}[z]$ and $\mathsf{E}_i = \mathbb{Q}[z]/(\Gamma_i)$ is a number field.

We assume throughout that $A \in \mathbb{Z}^{n\times n}$ with rank r and bottom $n-m$ rows all zero, whence $r \le m \le n$. B is specified by elements $\alpha_1, \ldots, \alpha_n$, $\beta_1, \ldots, \beta_n$, $\gamma_1, \ldots, \gamma_{2n-1} \in \mathsf{R}$ as

$$\begin{aligned} B &= D_1 T D_2 A \in \mathsf{R}^{n\times n}, \quad \text{where} \\ D_1 &= \operatorname{diag}(\alpha_1, \ldots, \alpha_n) \in \mathsf{R}^{n\times n}, \\ D_2 &= \operatorname{diag}(\beta_1, \ldots, \beta_n) \in \mathsf{R}^{n\times n}, \\ T &= \begin{pmatrix} \gamma_n & \gamma_{n+1} & \cdots & \gamma_1 \\ \vdots & \gamma_n & \ddots & \vdots \\ \gamma_{2n-2} & & \ddots & \gamma_{n+1} \\ \gamma_{2n-1} & \gamma_{2n-2} & \cdots & \gamma_n \end{pmatrix} \in \mathsf{R}^{n\times n}. \end{aligned} \tag{3.1}$$

In the notation of (1.1) and (1.2), $D_1 = \mathfrak{D}_1(\alpha_1, \ldots, \alpha_n)$, $D_2 = \mathfrak{D}_2(\beta_1, \ldots, \beta_n)$, $T = \mathfrak{T}(\gamma_1, \ldots, \gamma_{2n-1})$, and $B = \mathfrak{B}(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_{2n-1})$, where $\mathfrak{B}, \mathfrak{D}_1, \mathfrak{D}_2, \mathfrak{T} \in \mathbb{Z}[\Lambda]^{n\times n}$ and $\Lambda = \{v_1, \ldots, v_n, w_1, \ldots, w_n, y_1, \ldots, y_{2n-1}\}$ algebraically independent indeterminates over $\mathbb{Q}$.

The following lemma shows that for most choices of $\alpha_i, \beta_i, \gamma_i$ the characteristic polynomial is a power of x times the minimal polynomial.

LEMMA 3.1. *Let $A \in \mathbb{Z}^{n\times n}$ have rank r and $\mathfrak{B} = \mathfrak{D}_1\mathfrak{T}\mathfrak{D}_2 A \in \mathbb{Z}[\Lambda]^{n\times n}$ as in Theorem 1.2. Let*

$$\begin{aligned} \delta_0 &= \sum_{\sigma\in\mathcal{C}_r^n} (-1)^r \mathfrak{B}\binom{\sigma}{\sigma} \in \mathbb{Z}[\Lambda]\backslash\{0\}, \\ \delta_1 &= \operatorname{disc}(\operatorname{minpoly}(\mathfrak{B})) \in \mathbb{Z}[\Lambda]\backslash\{0\}, \\ \chi &= \delta_0 \cdot \delta_1 \in \mathbb{Z}[\Lambda]\backslash\{0\}. \end{aligned}$$

Let $\alpha_1, \ldots, \alpha_n$, $\beta_1, \ldots, \beta_n$, $\gamma_1, \ldots, \gamma_{2n-1}$ lie in an extension field E of $\mathbb{Q}$ and $B = D_1 T D_2 A \in \mathsf{E}^{n\times n}$ as in (3.1). Let $f = \operatorname{charpoly}(B) \in \mathsf{E}[x]$ and

$g = \text{minpoly}(B) \in \mathsf{E}[x]$. *With an abuse of notation we will write* $\chi(B)$ *for* $\chi(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_{2n-1})$.

(i) If $r = n$ *and* $\chi(B) \neq 0$ *then* $f = g$, *where* g *is squarefree,* $\deg g = n$ *and* $g(0) \neq 0$.

(ii) If $r < n$ *and* $\chi(B) \neq 0$ *then* $g = x\bar{g}$ *and* $f = x^{n-r+1} \cdot g$, *where* $\bar{g} \in \mathsf{E}[x]$ *is squarefree with* $\deg \bar{g} = r$ *and* $\bar{g}(0) \neq 0$. *It is always the case that* $g = x\bar{g}$ *for some (not necessarily squarefree)* $\bar{g}$ *with* $\deg \bar{g} \leq r$.

PROOF. Note that δ_0 is the coefficient on x^{n-r} of the characteristic polynomial of $\mathfrak{B}$. Both δ_0 and δ_1 are non-zero by Theorem 1.2. Assume $\chi(B) \neq 0$. Therefore the coefficient on x^{n-r} in the characteristic polynomial of $\mathfrak{B}$ is non-zero, so $\text{rank}\, B = r$. Also, every root of f in a splitting field, except perhaps zero, has multiplicity 1. If $r = n$ then, since g vanishes at each of these roots, $\deg g = n$ and $g(0) \neq 0$ since none of these roots is zero. If $r < n$, then f has r distinct roots aside from 0 and g must vanish at each of these, as well as zero. Since $\deg g \leq r + 1$, $g = x\bar{g}$. Since all eigenvalues of B have multiplicity 1 except for 0, $f = x^{n-r+1} \cdot g$. □

3.1. Modular reductions of R **and** B. We actually compute $\text{charpoly}(B) \in \mathsf{R}[x]$ by computing it modulo small primes and irreducible factors of $\Gamma \bmod p$ in $\mathbb{Z}_p[z]$, and then reconstruct the image in $\mathsf{R}[x]$ by the Chinese remainder algorithm. Care must be taken in the choice of these primes, since some may be "bad" in the sense that the structure of the problem may change locally. We proceed to bound from above the number of bad primes.

Let $\lambda \in \mathbb{R}$, $\mathcal{H} = \{-\lambda, \ldots, \lambda\} \subseteq \mathbb{Z}$ and $\mathcal{W} = \{h \bmod \Gamma : h \in \mathcal{H}[z], \deg h < \eta\} \subseteq \mathsf{R}$ as in `FindContents`. We also assume $\varrho = O(\log^2 m)$, $\eta < \varrho$, $\log \|\Gamma\| = O(\log^2 m)$ and $\log \lambda = O(\log m)$.

LEMMA 3.2. *Let* $\alpha_i, \beta_i, \gamma_j \in \mathcal{W}$ *for* $1 \leq i \leq n$, $1 \leq j \leq 2n - 1$ *and* B *as in* (3.1). *Then*

(i) $\|B\| \leq m\lambda^3\varrho^2 \cdot \|2\Gamma\|^{2\varrho} \cdot \|A\|$;

(ii) $\|\text{charpoly}(B)\| \leq n^r m^{2m} \lambda^{3m} \varrho^{3m} \cdot \|2\Gamma\|^{3m\varrho} \cdot \|A\|^m$.

PROOF. Part (i) follows easily from Lemma 2.4.

For part (ii), let $f = \text{charpoly}(B) = \sum_{0\le i\le n} a_i x^i$. Then, for $1 \le k \le r$,

$$a_{n-k} = (-1)^k \sum_{\sigma\in C_k^n} B\binom{\sigma}{\sigma},$$

$$\|B\binom{\sigma}{\sigma}\| < r^r \|B\|^r \varrho^{r-1} \|2\Gamma\|^{(r-1)\varrho},$$

$$\|a_{n-k}\| < n^r r^r m^r \lambda^{3r} \varrho^{3r-1} \cdot \|2\Gamma\|^{(3r-1)\varrho} \cdot \|A\|^r.$$

□

Since $\Gamma \in \mathbb{Z}[z]$ is squarefree, $\text{disc}(\Gamma) = \text{Res}_z(\Gamma, \partial\Gamma/\partial z) \in \mathbb{Z}\backslash\{0\}$. We say a prime p is *bad for* R if $\Gamma \bmod p$ is not squarefree in $\mathbb{Z}_p[z]$, i.e., such that $p \mid \text{disc}(\Gamma)$. The number of primes which are bad for R is at most

$$\begin{aligned}\log_2 |\,\text{disc}(\Gamma)| &\le \log_2 |\,\text{Res}_z(\Gamma, \partial\Gamma/\partial z)| \\ &\le \Upsilon := \log_2 \left((2\varrho-1)^{2\varrho-1} \cdot \|\Gamma\|^{\varrho-1} \cdot (\varrho\|\Gamma\|)^{\varrho}\right).\end{aligned}$$

Easily $\Upsilon = O(\log^4 m)$. When $p \nmid \text{disc}(\Gamma)$, $\Gamma \equiv \Gamma_1^{(p)} \cdots \Gamma_{k_p}^{(p)}$ for distinct, monic and irreducible $\Gamma_i^{(p)} \in \mathbb{Z}_p[z]$ and

$$\mathsf{R}/(p) = \mathsf{R}/(\Gamma_1^{(p)}, p) \oplus \mathsf{R}/(\Gamma_2^{(p)}, p) \oplus \cdots \oplus \mathsf{R}/(\Gamma_{k_p}^{(p)}, p), \tag{3.2}$$

a direct sum of finite fields $\mathsf{E}_i^{(p)} = \mathsf{R}/(\Gamma_i, p) \cong \text{GF}(p^{\deg \Gamma_i^{(p)}})$.

Next, let $g_i = \text{minpoly}(B \bmod \Gamma_i)$. Assume that either $r = n$ and g_i has degree n and $g_i(0) \ne 0$, or $r < n$, $g_i = x\bar{g}_i$, $\bar{g}_i \in \mathsf{E}_i[x]$ is squarefree of degree r and $\bar{g}_i(0) \ne 0$ for all $1 \le i \le k$. If B is specified as in Lemma 3.1, this lemma implies $\omega = \chi(B) \in \overline{\mathsf{R}}^*$. Let p be a prime dividing $\text{Res}_z(\chi(B))$. Such primes p are called *bad for* B, and there at most

$$\begin{aligned}\log_2 |\,\text{Res}_z(\omega, \Gamma)| &\le \log_2((2\varrho-1)^{2\varrho-1} \cdot \|\omega\|^{\varrho} \cdot \|\Gamma\|^{\varrho-1}) \\ &= O(\varrho\log\varrho + \varrho\log\|\omega\| + \varrho\log\|\Gamma\|) \\ &= O(\log\|\omega\| \cdot \log^2 m + \log^4 m).\end{aligned} \tag{3.3}$$

It remains only to bound $\log_2 \|\omega\|$. By Lemma 3.2,

$$\log \|\delta_0(\alpha_i, \beta_i, \gamma_i)\| \le \Delta_0 := \log_2 \left(n^r m^{2m} \lambda^{3m} \varrho^{3m} \cdot \|2\Gamma\|^{3m\varrho} \cdot \|A\|^m\right).$$

Easily, $\Delta_0 = O(m\log n + m\log^4 m + m\log\|A\|)$. Also, δ_1 is a $(2r-1)\times(2r-1)$ determinant of elements of this same magnitude, whence

$$\begin{aligned}\log_2\|\delta_1(\alpha_i, \beta_i, \gamma_i)\| &\le \log_2 \left((2r-1)^{2r-1} r^r \|B\|^{2r-1} \varrho^{2r-2} \cdot \|2\Gamma\|^{\varrho(2r-2)}\right) \\ &\le \Delta_1 := \log_2 \left((2r)^{2r} n^{2r^2} r^r m^{2mr} \lambda^{3mr} \varrho^{3mr+2r} \cdot \|A\|^{mr} \cdot \|2\Gamma\|^{(3mr+2r)\varrho}\right).\end{aligned}$$

Clearly, $\Delta_1 = O(m^2 \log n + m^2 \log^4 m + m^2 \log \|A\|)$. Thus

$$\log_2 \|\omega\| \leq \Delta_0 + \Delta_1 + \log_2(\varrho \cdot \|2\Gamma\|^{\varrho}) = O(m^2 \log n + m^2 \log^4 m + m^2 \log \|A\|)$$

and there are at most

$$\begin{aligned}\Delta :=& 2\varrho \log_2(2\varrho) + \varrho \log_2 \|\Gamma\| + \varrho \cdot (\Delta_0 + \Delta_1 + \log_2(\varrho \cdot \|2\Gamma\|^{\varrho})) \\ =& O(m^2 \log(n) \log^2 m + m^2 \log^6 m + m^2 \log^2 m \log \|A\|)\end{aligned}$$

bad primes for B using (3.3).

3.2. Characteristic polynomial via modular minimal polynomial. Let p be a prime which is *good* (i.e. not bad) for both R and B. For such a p, $B_i^{(p)} = B \bmod (\Gamma_i^{(p)}, p)$ is an $n \times n$ matrix over the finite field $\mathsf{E}_i^{(p)}$ and $g_i^{(p)} = \text{minpoly}(B_i^{(p)}) \equiv g_i \bmod p$, for $1 \leq i \leq k_p$.

We compute $f = \text{charpoly}(B) \in \mathsf{R}[x]$ modulo sufficiently many good primes to recover it in $\mathsf{R}[x]$. The product of good primes required is at least $2\|f\| \leq \Psi := 2n^r m^{2m} \lambda^{3m} \varrho^{3m} \|2\Gamma\|^{3m\varrho} \cdot \|A\|^m$. By Rosser & Schoenfeld (1962),

$$\begin{aligned} 0.34x/\log(x) < \pi(x) &= \sum_{\substack{p \leq x \\ p \text{ prime}}} 1 < 1.26x/\log(x), \\ x/3 < \vartheta(x) &= \sum_{\substack{p \leq x \\ p \text{ prime}}} \log p. \end{aligned} \tag{3.4}$$

Thus, $\vartheta(3 \log \Psi) > \log \Psi$ and $\pi(3\Psi) < 4 \log(\Psi)/\log(\log(\Psi))$ is an upper bound on the number of good primes required. We select primes randomly from a set of primes $\mathcal{P} \subseteq \mathbb{N}$ consisting of primes p greater than $4m + 3$ (for reasons discussed below) and such that $\mathbb{Z}_p$ supports a $2n$-point Fast Fourier Transform (FFT). This latter condition is equivalent to $\mathbb{Z}_p$ having a 2^lth primitive root of unity where $l = \lfloor 1 + \log_2(2n) \rfloor$ or that $p \equiv 1 \bmod 2^l$. It allows us to multiply two polynomials of degree less than n over an algebraic extension field of $\mathbb{Z}_p$ with $O(n \log n)$ operations in that field. We call such a prime an n-Fourier prime. We require

$$\#\mathcal{P} \geq \max\{2\Upsilon + 2\Delta, 4 \log(\Psi)/\log(\log(\Psi))\}$$

to ensure good primes are found with high probability. Let $\pi_l(x)$ be the number of primes p with $p \leq x$ and $p \equiv 1 \bmod 2^l$. It was shown by Dirichlet (see, e.g., Gelfond & Linnik 1965) that

$$\pi_l(x) = x/(2^{l-1} \log x) + o(x/(2^{l-1} \log x)). \tag{3.5}$$

Thus

$$\pi_l(n\log(n)\cdot(\#\mathcal{P})\cdot\log(\#\mathcal{P})) - \pi_l(4m+3) = \Omega(\#\mathcal{P})$$

and we may assume the primes used in $\mathcal{P}$ have $O(\log(n\log(n)\cdot(\#\mathcal{P})\cdot\log(\#\mathcal{P})))$ or $O(\log n + \log\log\|A\|)$ bits. A simple sieving algorithm can be used to compute $\mathcal{P}$ using the smallest primes possible. At this stage we may also factor Γ mod p for each $p \in \mathcal{P}$ using Berlekamp's algorithm, and eliminate from $\mathcal{P}$ those p modulo which Γ is not squarefree.

We now give an algorithm which reduces the problem of computing the characteristic polynomial $f \in \mathsf{R}[x]$ of $B \in \mathsf{R}^{n\times n}$ to computing the characteristic polynomial of a homomorphic image of B in a finite field.

Algorithm: `CharpolyViaModMinpoly`

Input: – $B = D_1TD_2A \in \mathsf{R}^{n\times n}$ (represented explicitly or as a black box) as above;
– $r = \operatorname{rank} A$;

Output: – $f = \operatorname{charpoly}(B) \in \mathsf{R}[x]$ or a report that "$\chi(B) = 0$";
If f is returned, the output is always correct;
If "$\chi(B) = 0$" is reported, it is correct with probability $> 1/2$;

(1) Construct a set $\mathcal{P}$ of n-Fourier primes as above, with $\#\mathcal{P} \geq 4\log(\Psi)/\log\log(\Psi)$ and such that Γ mod p is squarefree for all $p \in \mathcal{P}$;
`Attempts`:= 0; `BisGood`:=`false`; `GoodPrimes`:= {};

(2) While #`GoodPrimes`$< 4\log(\Psi)/\log\log(\Psi)$ and (`Attempts`< 3 or `BisGood`) Do

(3) Choose a random prime $p \in \mathcal{P}$;
Assume $\Gamma \equiv \Gamma_1^{(p)}\cdots\Gamma_{k_p}^{(p)} \bmod p$ for distinct, irreducible $\Gamma_1^{(p)},\ldots,\Gamma_{k_p}^{(p)} \in \mathbb{Z}_p[x]$;
`Attempts` := `Attempts` +1;

(4) For `i` from 1 to k_p Do

(5) Compute $g_i^{(p)} = \operatorname{minpoly}(B \bmod (\Gamma_i^{(p)}, p))$;
{ may return a proper factor of $g_i^{(p)}$ with probability at most $1/2$ }

(6) If $r = n = \deg g_i^{(p)}$ and $g_i^{(p)}(0) \neq 0$ Then $f_i^{(p)} := g_i^{(p)}$;
Else If $r < n$ and $r = 1 + \deg g_i^{(p)}$ Then $f_i^{(p)} := x^{n-r+1}\cdot g_i^{(p)}$;
Else Goto (2); { next iteration of While loop }
EndFor;

(7) Construct $f^{(p)} = f \bmod p$ from $f_1^{(p)},\ldots,f_{k_p}^{(p)}$ using the Chinese remainder algorithm;

(8) `GoodPrimes` := `GoodPrimes` $\cup\{p\}$; `BisGood` := `true`;
EndWhile;
(9) If `BisGood`
Construct $f \in \mathsf{R}[x]$ from $f^{(p)}$ for $p \in$`GoodPrimes` using the Chinese remainder algorithm;
Return f;
Else Return "failure";

THEOREM 3.3. *The algorithm* `CharpolyViaModMinpoly` *works as stated. It requires an expected number of* $O^{\sim}(m \log \|A\|)$ *computations of the minimal polynomial of* $B \bmod (\Gamma_i^{(p)}, p)$ *for* n*-Fourier primes* p *with* $O(\log n + \log\log \|A\|)$ *bits and all irreducible factors* $\Gamma_i^{(p)} \in \mathbb{Z}_p[z]$ *of* $\Gamma \bmod p$. *An additional* $O^{\sim}(m^3 \log^2 \|A\|)$ *bit operations and storage for* $O^{\sim}(m^2 \log \|A\|)$ *bits are sufficient.*

PROOF. The algorithm computes the minimal polynomial (via a subroutine described below) modulo randomly chosen primes $p \in \mathcal{P}$ and irreducible factors of $\Gamma \bmod p$. If $r = n = \deg g_i^{(p)}$ and $g_i^{(p)}(0) \neq 0$, or $r < n$ and $r = 1 + \deg g_i^{(p)}$, then we can identify a correct homomorphic image of $f \bmod (p, \Gamma_i^{(p)})$ by Lemma 3.1. If we can do this for all factors of $\Gamma \bmod p$ we can construct a correct homomorphic image of $f \bmod p$. If we are able to do this for any one prime we have demonstrated that $\chi(B) \neq 0$ (and note this by setting `BisGood` to `true`).

Finally, if $\chi(B) \neq 0$, the probability that `BisGood` is set true in each iteration of the loop (3) - (8) is at most $1/4$, since at most half of the primes in $\mathcal{P}$ are bad for B, and the minimal polynomial computation fails with probability at most $1/2$. With 3 iterations we correctly identify $\chi(B) \neq 0$ with probability at least $1-(3/4)^3 > 1/2$. Once a good prime is found, we are confident to continue to choose random primes and compute homomorphic images of f until sufficiently many are found to recover $f \in \mathsf{R}[x]$. This requires an expected number of iterations at most double the number primes required. The expected number of primes p examined is then less than $8\log(\Psi)/\log\log(\Psi) = O^{\sim}(m \log \|A\|)$ and we must compute $\text{minpoly}(B \bmod (\Gamma_i^{(p)}, p))$ for $1 \leq i \leq k_p$ for each such p.

The cost in addition to the modular minimal polynomial computations is that of the modular reductions and Chinese remainder computations. The latter cost dominates and requires that we recover the $r \leq m$ non-zero coefficients of f in R, each of which is a polynomial in $\mathbb{Z}[z]$ of degree $\varrho = O(\log^2 m)$. These coefficients are represented modulo a collection of primes with $O(\log n + \log\log \|A\|)$ bits each, and product at least Ψ. The Chinese remainder algorithm on such input requires $O(\log^2 \Psi)$ or $O^{\sim}(m^2 \log^2 \|A\|)$ bit operations and there are at most m non-zero coefficients to recover. □

We next describe two methods to compute the minimal polynomial of a matrix, one for sparse matrices and one for dense matrices.

LEMMA 3.4. *Let $A \in \mathbb{Z}^{n\times n}$ have bottom $n-m$ rows zero. Let K be a finite field of characteristic p with at least $4m+3$ elements, where p is an n-Fourier prime. Define matrices $\bar{U} = \bar{D}_1\bar{T}\bar{D}_2 \in \mathsf{K}^{n\times n}$, where $\bar{T} \in \mathsf{K}^{n\times n}$ a Toeplitz matrix and $\bar{D}_1, \bar{D}_2 \in \mathsf{K}^{n\times n}$ diagonal matrices and $\bar{A} \in \mathsf{K}^{n\times n}$, the embedding of A into $\mathbb{Z}_p^{n\times n}$. We can compute the minimal polynomial of $\bar{B} = \bar{U}\bar{A}$ with either*

(i) $3m$ matrix-vector products $w \to \bar{A}w$ for $w \in \mathsf{K}^{n\times 1}$ and an additional $O(nm)$ operations in K, or

(ii) $O((n/m)\cdot \mathrm{MM}(m)\log m)$ operations in $\mathsf{K}^{n\times 1}$.

In both cases, the computed minimal polynomial is correct with probability at least $1/2$, and a proper factor otherwise.

PROOF. For part (i), Kaltofen & Saunders (1991) show that we can accomplish this with $3r$ matrix-vector products and an additional $O(nr)$ operations in K (see also Wiedemann 1986). Their algorithm returns the correct minpoly$(\bar{B})$ with probability at least $(1-(r+1)/(4r+3))^2 > 1/2$, and a proper factor of it otherwise.

For part (ii) for dense matrices we use a method similar to that of Wiedemann (1986) for sparse matrices. Choose random $u \in \mathsf{K}^{1\times n}$, $v \in \mathsf{K}^{n\times 1}$ and compute the first $2m$ terms in the linearly recurring sequence

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

Wiedemann (1986) shows that the minimal polynomial of this sequence is the minimal polynomial of $\bar{B}$ with probability at least $1-((r+1)/(4r+3))^2 > 1/2$. To accomplish this we compute $\bar{B}^iv$ for $0 \le i \le 2m-1$. Assume by the beginning of stage j we have computed $U_j = (\bar{A}\bar{T})^{2^j}$ and $\bar{B}^iv$ for $0 \le i < 2^j$. During stage j compute

$$(\bar{T}U_jA)[v|\bar{B}v|\cdots|\bar{B}^{2^j-1}v] = [\bar{B}^{2^j+1}v|\bar{B}^{2^j+2}v|\cdots|\bar{B}^{2^{j+1}}v]$$

and U_j^2. Since U_j and U_jA have bottom $n-m$ rows zero, these can both be accomplished with $O((n/m)\,\mathrm{MM}(m))$ operations in K. The total cost is then $O((n/m)\,\mathrm{MM}(m)\log m)$ operations in K. Using the Berlekamp-Massey algorithm we can compute the minimal polynomial of this sequence with $O(m^2)$ operations in K. □

COROLLARY 3.5 *(to Theorem 3.3). Let $A \in \mathbb{Z}^{n\times n}$ with bottom $n-m$ rows zero and $B \in \mathsf{R}^{n\times n}$ as in (3.1). The algorithm* `CharpolyViaModMinpoly` *either computes (correctly) the characteristic polynomial of B, or reports that $\chi(B) = 0$ (correctly with probability at least $1/2$). Depending upon the implementation of the modular minimal polynomial algorithm chosen, it requires either*

(i) $O^{\sim}(m^2 \log \|A\|)$ evaluations of $v \mapsto Av \bmod p$, where p is a prime with $O(\log n + \log\log \|A\|)$ bits and $v \in \mathbb{Z}_p^{n\times 1}$, and an additional $O^{\sim}(nm^2 \log \|A\| + m^3 \log^2 \|A\|)$ bit operations and storage for $O^{\sim}(n + m^2 \log \|A\|)$ bits, or

(ii) $O^{\sim}(n \cdot \mathrm{MM}(m) \cdot \log \|A\| + m^3 \log^2 \|A\|)$ bit operations and storage for $O^{\sim}(nm + m^2 \log \|A\|)$ bits.

PROOF. Let p be an n-Fourier prime and $1 \le i \le k_p$. Let $\wp = \mathsf{R}\Gamma_i + \mathsf{R}p$, a prime ideal in R and $\mathsf{E}_i^{(p)} = \mathsf{R}/\wp$ as in (3.2).

For case (i) we first look at the cost of computing $Bv \bmod \wp$ for $v \in (\mathsf{E}_i^{(p)})^{n\times 1}$. For $v \in \mathsf{E}_i^{(p)}$ we can compute $Av \bmod \wp$ with $O(\deg \Gamma_i^{(p)})$ evaluations $w \mapsto (Aw \bmod p)$ for $w \in \mathbb{Z}_p^{n\times 1}$.

Recall that $B = D_1 T D_2 A$. To compute $Tv \bmod \wp$ we use the fact that multiplication of a vector by a Toeplitz matrix is equivalent to two multiplications of polynomials of degree n. We can therefore compute $Tv \bmod \wp$ with $O(n \log n)$ operations in $\mathsf{E}_i^{(p)}$ using an FFT. Each operations in $\mathsf{E}_i^{(p)}$ requires $O(\deg(\Gamma_i^{(p)}) \log \deg(\Gamma_i^{(p)}))$ operations in $\mathbb{Z}_p$. It is easily seen that the most expensive case is when $k_p = 1$ and $\Gamma \bmod p$ is irreducible in $\mathbb{Z}_p[z]$. Since $\deg(\Gamma) = O(\log^2 m)$, the cost to evaluate $Bv \bmod \wp$ is $O^{\sim}(n)$ bit operations plus the $O(\deg \Gamma)$ evaluations of $A \bmod p$. The cost to find the minimal polynomial of $B \bmod \wp$ is therefore $O^{\sim}(nm)$ bit operations plus the $O^{\sim}(m)$ black box evaluations of $A \bmod p$ using Lemma 3.4(i). By Theorem 3.3, the total cost of `CharpolyViaModMinPoly` using Lemma 3.4(i) is $O^{\sim}(nm^2 \log \|A\| + m^3 \log^2 \|A\|)$ bit operations.

For case (ii), we also need to compute the minimal polynomial of $B \bmod \wp$. Again it is easily seen that the most expensive case is when $k_p = 1$ and $\Gamma \bmod p$ is irreducible in $\mathbb{Z}_p[z]$. We can therefore find the minimal polynomial of $B \bmod \wp$ with $O^{\sim}((n/m) \cdot \mathrm{MM}(m))$ bit operations using Lemma 3.4(ii). By Theorem 3.3, the total cost of `CharpolyViaModMinPoly` using the algorithm described in Lemma 3.4(ii) is $O^{\sim}(n\,\mathrm{MM}(m) \log \|A\| + m^3 \log^2 \|A\|)$ bit operations. Additional storage for $O^{\sim}(nm + m^2 \log \|A\|)$ bits is sufficient. □

4. Computing the Smith form

We complete our Smith form algorithm by applying the algorithms of Section 3 for computing the characteristic polynomial of the special matrix $B \in \mathsf{R}^{n\times n}$, to the algorithm `FindContents` of Section 2.

It is useful to determine the rank of A in advance. This could accomplished "on the fly" from the characteristic polynomial of B, but the algorithms of Wiedemann (1986) and Kaltofen & Saunders (1991) are simpler.

LEMMA 4.1. *Let $A \in \mathbb{Z}^{n\times n}$ with bottom $n-m$ rows zero and rank $r \le m \le n$. We can determine the rank of A with*

(i) $O(m\log(1/\epsilon))$ matrix-vector products $w \mapsto (Aw \bmod p)$ for a prime p with $\log p = O(\log n + \log\log\|A\|)$, $w \in \mathbb{Z}_p^{n\times 1}$, and $\tilde{O}(nm\log(1/\epsilon))$ additional bit operations.

(ii) $\tilde{O}((n/m)\cdot \mathrm{MM}(m)\log(1/\epsilon) + nm\log\|A\|\log(1/\epsilon))$ bit operations.

PROOF. We rely on the fact that $\mathrm{rank}(A \bmod p) = \mathrm{rank}\, A$ if and only if $p \nmid d_r$. Since $d_r < b := n^m\|A\|$, it is divisible by at most $4\log(b)/\log\log(b)$ distinct primes by (3.4). Construct a set of $8\log(b)/\log\log(b)$ small primes $\mathcal{Q} \subseteq \mathbb{N}$ as follows. We assume that every of $p \in \mathcal{Q}$ is at least $3n(n+1)$. We also insist that $p \equiv 1 \bmod 2^l$, for $l = \lfloor 1 + \log(2n)\rfloor$ so that two polynomials of degree n over $\mathbb{Z}_p$ can be multiplied with $O(n\log n)$ operations in $\mathbb{Z}_p$. Let $\mathcal{Q}$ be the set of the smallest such primes; $\mathcal{Q}$'s entries have size $O(\log(b)\cdot n\log n)$ and hence $O(\log n + \log\log\|A\|)$ bits.

For part *(i)*, choose primes p randomly from $\mathcal{Q}$ and compute the rank of $A \bmod p$. For each prime chosen from $\mathcal{Q}$, $\mathrm{rank}(A) = \mathrm{rank}(A \bmod p)$ with probability at least $1/2$. Using the algorithm of Kaltofen & Saunders (1991) we obtain the correct rank of $A \bmod p$ with probability at least $1/2$ (and a smaller rank otherwise), with $\tilde{O}(nm)$ bit operations and $O(m)$ matrix vector products by $(A \bmod p)$. Repeating this for $2.5\log_2(1/\epsilon)$ primes, we can guarantee that the maximum rank found is the correct rank of A with probability at least $1-\epsilon$.

For part *(ii)*, we again choose a random prime $p \in \mathcal{Q}$ and reduce $A \bmod p$. We then use the asymptotically fast algorithm of Ibarra *et al.* (1982) to compute the rank of $A \bmod p$ (correctly) using $O((n/m)\cdot \mathrm{MM}(m))$ operations in $\mathbb{Z}_p$. With $\log_2(1/\epsilon)$ choices of primes $p \in \mathcal{Q}$, the maximum rank obtained is the rank of A with probability at least $1-\epsilon$. □

We are now ready to apply the algorithm `FindContents` to determine the determinantal divisors $d_1, \ldots, d_r$, and hence the Smith normal form, of A. For $1 \le i \le r$, let f_k be the coefficient on x^{n-k} of the characteristic polynomial of

$\mathfrak{B} \in \mathbb{Z}[\Lambda]^{n\times n}$ as in Theorem 1.2, where it is shown that $d_k = \mathrm{cont}(f_k) \in \mathbb{Z}$. In algorithm `CharpolyViaModMinpoly` and Corollary 3.5*(i)* we showed how to compute the characteristic polynomial of $\mathfrak{B}$ for a sparse matrix A.

THEOREM 4.2. *Let $A \in \mathbb{Z}^{n\times n}$ with bottom $n-m$ rows zero and $\epsilon > 0$. We can compute the Smith form $S \in \mathbb{Z}^{n\times n}$ of A with either*

(i) $O^{\sim}(m^2 \log \|A\| \cdot \log(1/\epsilon))$ evaluations of $Av \bmod p$, where p is a prime with $O(\log n + \log\log \|A\|)$ bits and $v \in \mathbb{Z}_p^{n\times 1}$; an additional $O^{\sim}((nm^2 \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ bit operations and storage for $O^{\sim}(n + m^2 \log \|A\|)$ bits is also required; or

(ii) $O^{\sim}((n\,\mathrm{MM}(m) \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ bit operations and additional storage for $O^{\sim}(nm + m^2 \log \|A\|)$ bits.

The output is correct with probability at least $1 - \epsilon$.

PROOF. First compute the rank of A using the methods described in Lemma 1.1. Compute it correctly with probability at least $1 - 1/(2\epsilon)$.

Next use `FindContents` to determine the contents $d_1, \ldots, d_r$ of $f_1, \ldots, f_r$ respectively. For part *(i)* employ the algorithm `CharPolyViaModMinpoly` with Wiedemann's algorithm for the minimal polynomial to compute the necessary modular characteristic polynomials described in Corollary 3.5*(i)*. For part *(ii)* use the asymptotically fast minimal polynomial computation for (dense) matrices with $n - m$ rows of zeros, as described in Corollary 3.5*(ii)*. We apply Theorem 2.9 with error tolerance $1/(2\epsilon)$ to obtain the cost. Note that $\|f_i\| \le n^r \|A\|^r$, whence $O(\log(1/\epsilon) + \log m + \log\log n + \log\log \|A\|)$ characteristic polynomial computations are required. Computations are in the ring R, where $\varrho = \deg \Gamma = O(\log^2 m)$, $\log \|\Gamma\| = O(\log^2 m)$ and inputs to the black box have height $r^{O(1)}$.

For *(i)*, Corollary 3.5*(i)* shows this requires $O^{\sim}((nm^2 \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ additional bit operations. Additional storage for $O^{\sim}(n + m^2 \log \|A\|)$ bits is sufficient.

For *(ii)*, Corollary 3.5*(ii)* shows this requires $O^{\sim}((n\,\mathrm{MM}(m) \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ additional bit operations. Additional storage for $O^{\sim}(nm + m^2 \log \|A\|)$ bits is sufficient.

We return $S = \mathrm{diag}(d_1, d_2/d_1, \ldots, d_r/d_{r-1}, 0, \ldots, 0)$. □

Acknowledgements

Research was supported in part by Natural Sciences and Engineering Research Council of Canada research grant OGP0155376, and University of Manitoba research grant 431-1725-80.

References

E. R. Berlekamp, Factoring polynomials over large finite fields. *Math. Comp.* **24** (1970), 713–735.

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** (1982), 687–708.

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

A. O. Gelfond and Yu. B. Linnik, *Elementary Methods in Analytic Number Theorey*. George Allen & Unwin Ltd., London, 1965.

M. Giesbrecht, *Nearly Optimal Algorithms for Canonical Matrix Forms*. PhD thesis, University of Toronto, 1993. 196 pp.

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

M. Giesbrecht, Probabilistic computation of the Smith normal form of a sparse integer matrix. In *Algorithmic Number Theory: Second International Symposium*, ed. H. Cohen, 1996, 175–188. Proceedings to appear in Springer's Lecture Notes in Computer Science.

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

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

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

O. Ibarra, S. Moran, and R. Hui, A generalization of the fast LUP matrix decomposition algorithm and application. *J. of Algorithms* **3** (1982), 45–56.

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** (1989), 658–669.

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.

E. KALTOFEN, M. S. KRISHNAMOORTHY, AND B. D. SAUNDERS, Fast parallel computation of Hermite and Smith forms of polynomial matrices. *SIAM J. Algebraic and Discrete Methods* **8** (1987), 683–690.

E. KALTOFEN, M. S. KRISHNAMOORTHY, AND B. D. SAUNDERS, Parallel algorithms for matrix normal forms. *Linear Algebra and its Applications* **136** (1990), 189–208.

R. KANNAN AND A. BACHEM, Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comp.* **8** (1979), 499–507.

W. KELLER-GEHRIG, Fast algorithms for the characteristic polynomial. *Theor. Computer Science* **36** (1985), 309–317.

R. LIDL AND H. NIEDERREITER, *Finite Fields*, vol. 20 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, Reading MA, 1983.

M. NEWMAN, *Integral Matrices*. Academic Press, New York, 1972.

J. B. ROSSER AND L. SCHOENFELD, Approximate formulas for some functions of prime numbers. *Ill. J. Math.* **6** (1962), 64–94.

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

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

A. STORJOHANN, 1995. Personal Communication.

A. STORJOHANN, Near optimal algorithms for computing smith normal forms of integer matrices. In *Proceedings of ISSAC'96*, Zurich, Switzerland, 1996, 267–274.

W.D. WALLIS, A.P. STREET, AND J. SEBERRY WALLIS, *Combinatorics: room squares, sum-free sets, Hadamard matrices*, vol. 292 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1972.

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

R. ZIPPEL. Probabilistic algorithms for sparse polynomials. In *Proc. EUROSAM 79*. Marseille, 1979, 216–226.

Manuscript received October 20, 1996

MARK GIESBRECHT
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada, R3T 2N2
mwg@cs.umanitoba.ca

Fast Computation of the Smith Normal Form of an Integer Matrix

Mark Giesbrecht†
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba
Canada, R3T 3T6
mwg@cs.umanitoba.ca

Abstract

We present two new probabilistic algorithms for computing the Smith normal form of an $A \in \mathbb{Z}^{m\times n}$. The first requires an expected number of $O(m^2 n \cdot \mathrm{M}(m \log \|A\|))$ bit operations (ignoring logarithmic factors) and is of the Las Vegas type; that is, it never produces an incorrect answer. Here $\|A\| = \max_{ij} |A_{ij}|$ and $\mathrm{M}(l)$ bit operations are sufficient to multiply two l-bit integers ($\mathrm{M}(l) = l^2$ using standard arithmetic). This improves on the previously best known (deterministic) algorithm of Hafner and McCurley, which requires about $O(m^3 n \log \|A\| \cdot \mathrm{M}(m \log \|A\|))$ bit operations. We also present an even faster, more space efficient algorithm which requires an expected number of $O((m^3 n \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$ bit operations *using standard integer arithmetic*. This algorithm is of the Monte Carlo type: it returns the correct result with probability at least $1-\epsilon$ for a user specified tolerance $\epsilon > 0$. This algorithm also requires only $O(nm \log \|A\|)$ bits of storage, versus $O(nm^2 \log \|A\|)$ bits required by other known algorithms.

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

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

where $r = \mathrm{rank}(A)$ 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.

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

Appears in Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC'95

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 this paper we present two new probabilistic algorithms to compute the Smith normal form of an integer matrix. Their costs are substantially smaller than the previously best known (deterministic) algorithm of Hafner & McCurley (1989,1991; see also Kannan & Bachem 1979, Chou & Collins 1982, Domich *et al.* 1987, Iliopolous 1989).

Suppose $A \in \mathbb{Z}^{m\times n}$ has rank r. Without loss of generality we assume throughout that $m \le n$ — the Smith normal form is invariant under transpose. In Section 1 we present a fast new probabilistic algorithm to compute the Smith normal form of A. It requires an expected number of $O^{\sim}(m^2 n \cdot \mathrm{M}(m \log \|A\|))$ bit operations and is of the Las Vegas type, that is, it always produces the correct answer. For convenience we use "soft-Oh" notation: for any $f, g : \mathbb{R}^s \to \mathbb{R}$, $f = O^{\sim}(g)$ if and only if $f = O(g \cdot \log^c g)$ for some constant $c > 0$. The deterministic algorithm of Hafner & McCurley (1991) requires $O^{\sim}(m^3 n \log \|A\| \cdot \mathrm{M}(m \log \|A\|))$ bit operations in the worst case. Like many previous algorithms, ours computes modulo a multiple d of the non-zero invariant factors of A. It requires an expected number of $O(rmn \log\log d)$ operations in $\mathbb{Z}_d$. Such a d is easily found, but may contain $O(m(\log \|A\| + \log m))$ bits. Hafner & McCurley's (1991) algorithm also computes modulo d, but requires $O(rmn \log d)$ operations in $\mathbb{Z}_d$. Our algorithm is related to Hafner & McCurley's, but with a random, unimodular column operation before each phase of row reduction. The worst case cost of Hafner & McCurley's algorithm may well be encountered only on a small portion of inputs, but our randomization avoids it on all inputs with high probability.

In Section 2 we present an even faster probabilistic algorithm for the Smith normal form of $A \in \mathbb{Z}^{m\times n}$ which is considerably more space efficient as well. Define θ such that we can multiply two $r \times r$ matrices over a ring R with $O(r^\theta)$ operations in R; using standard matrix arithmetic $\theta = 3$, while the best know algorithm of Coppersmith & Winograd (1990) allows $\theta = 2.38$. This new algorithm to compute the Smith normal form of $A \in \mathbb{Z}^{m\times n}$ then requires an expected number of

$$O^{\sim}((m^\theta n \log \|A\| + m^3 \log^2 \|A\|) \cdot \log(1/\epsilon))$$

bit operations *using standard integer arithmetic*. This al-

gorithm 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 $1-\epsilon$. By comparison Hafner & McCurley's (1991) algorithm requires $\tilde{O}(m^5 n \log^3 \|A\|)$ bit operations using standard integer arithmetic (and $\tilde{O}(m^4 n \log^2 \|A\|)$ bit operations using currently impractical but asymptotically fast arithmetic). Moreover, this new algorithm requires additional storage for only $r = \text{rank}(A)$ "large" integers, each with $\tilde{O}(r \log \|A\|)$ bits. These integers converge on the determinantal divisors of A as the algorithm proceeds. Previous algorithms, which work with dense $m \times n$ matrices in $\mathbb{Z}_d$, require $O(nm \log d)$ or $\tilde{O}(nm^2 \log \|A\|)$ bits of storage.

Our Monte Carlo algorithm is more akin to the methods of Kaltofen *et al.* (1987) for computing Smith normal forms of matrices of polynomials than to the modulo-determinant algorithms discussed. Hafner & McCurley (1991) conjectured that these methods might be adapted to work for integer matrices. Our algorithm demonstrates this, showing that we can find the Smith normal form of an integer matrix with about the same order of cost as is required to compute the determinant of that matrix. Additional memory required is on the order of the size of the input and output.

In neither the Las Vegas modulo-determinant algorithm of Section 1 nor the Monte Carlo algorithm of Section 2 do we compute the transition matrices to the Smith normal form. 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. However, many applications of Smith normal form computations, such as determining the structure of the class group of a number field, do not require the transition matrices be computed.

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 $B \in \mathsf{P}^{m \times n}$, $(b_1, \ldots, b_k) \in \mathcal{C}_k^m$ and $(c_1, \ldots, c_k) \in \mathcal{C}_k^n$ define the $(k \times k)$ *minor* of a B:

$$B\begin{pmatrix} b_1 \cdots b_k \\ c_1 \cdots c_k \end{pmatrix} = \det \begin{pmatrix} B_{b_1 c_1} & B_{b_1 c_2} & \cdots & B_{b_1 c_k} \\ B_{b_2 c_1} & B_{b_2 c_2} & \cdots & B_{b_2 c_k} \\ \vdots & & & \vdots \\ B_{b_k c_1} & B_{b_k c_2} & \cdots & B_{b_k c_k} \end{pmatrix} \in \mathsf{P}.$$

The *kth determinantal divisor* $d_k \in \mathsf{P}$ *of* B is defined as the GCD of all $k \times k$ minors of B. A well known definition of the invariant factors of B is as a quotient of determinantal divisors of B. If B has Smith normal form $S = \text{diag}(s_1, \ldots, s_r, 0, \ldots, 0) \in \mathsf{P}^{m \times n}$ then $s_1 = d_1$ and $s_i = d_i / d_{i-1}$ for $2 \le i \le r$. Thus if we can find all the determinantal divisors of B, we can easily recover the invariant factors of B.

1 A Las Vegas Algorithm for the Smith normal form

In this section we present a fast Las Vegas type probabilistic algorithm for computing the Smith normal form of an $A \in \mathbb{Z}^{m \times n}$ of rank r and $m \le n$. Like many previous algorithms (see Hafner & McCurley 1991), this algorithm computes in $\mathbb{Z}_d$, where d is a multiple of the non-zero invariant factors of A. Our new algorithm requires an expected $O(mnr \log \log d)$ operations in $\mathbb{Z}_d$, and *always* produces the correct answer. A suitable d with $\lceil \log_2 d \rceil = O(m(\log m + \log \|A\|))$ bits can be found quickly, yielding an algorithm which costs $\tilde{O}(m^2 n \cdot \mathrm{M}(m \log \|A\|))$ bit operations.

The basis for modulo-determinant computation of the Smith normal form of A is as follows (see Hafner & McCurley 1989, Section 3). Let $A \in \mathbb{Z}^{m \times n}$ have rank r and Smith form $S = \text{diag}(s_1, \ldots, s_r, 0, \ldots, 0) \in \mathbb{Z}^{m \times n}$. Let d be a multiple of $s_1 \cdots s_r$ and $\bar{A} = A \bmod d$. If $\bar{S} = \text{diag}(\bar{s}_1, \ldots, \bar{s}_r, 0, \ldots, 0) \in \mathbb{Z}_d^{m \times n}$ is the Smith normal form of $\bar{A}$ over $\mathbb{Z}_d$, then $d_i = \gcd(\bar{s}_1 \cdots \bar{s}_i, d) \in \mathbb{Z}$ is the ith invariant factor of A for $1 \le i \le r$ and $s_1 = d_1$, $s_i = d_i / d_{i-1}$ for $2 \le i \le r$.

We initially count operations in $\mathbb{Z}_d$. For $a, b \in \mathbb{Z}_d$ we can find $a + b, a - b, ab \in \mathbb{Z}_d$, and determine if $a \mid b$ (i.e., $a = bc$ for some $c \in \mathbb{Z}_d$) with $O(\mathrm{M}(\log d) \log \log d)$ bit operations. In this same time we can also find

$$\gcd_d(a, b) = \left\{ \begin{array}{c} \gcd(d, \bar{a}, \bar{b}) \bmod d : \bar{a}, \bar{b} \in \mathbb{Z} \\ \bar{a} \equiv a \bmod d, \bar{b} \equiv b \bmod d \end{array} \right\} \in \mathbb{Z}_d,$$

the "least" principal generator of the ideal $(a, b) \subseteq \mathbb{Z}_d$.

Fact 1.1 (Hafner & McCurley 1991) *Let $\bar{A} \in \mathbb{Z}_d^{m \times n}$ be such that the first column is not all zeros. We can find a $\bar{P} \in \mathbb{Z}_d^{m \times m}$ with $\det \bar{P} = \pm 1$ such that $\bar{B} := \bar{P}\bar{A} \in \mathbb{Z}_d^{m \times n}$ satisfies $\bar{B}_{11} = \gcd_d(\bar{A}_{11}, \ldots, \bar{A}_{m1})$ and $\bar{B}_{i1} = 0$ for $2 \le i \le m$. The algorithm requires $O(nm)$ operations in $\mathbb{Z}_d$.*

Lemma 1.2 *Assume $\bar{A} \in \mathbb{Z}_d^{m \times n}$ and $\bar{v} = (1, v_2, \ldots, v_n) \in \mathbb{Z}_d^{n \times 1}$ is chosen randomly and uniformly from $\mathbb{Z}_d$. If $w = (w_1, \ldots, w_m)^t := \bar{A}\bar{v}$ then*

$$\begin{aligned} \text{Prob}\{\gcd_d(\, & w_1, \ldots, w_m) \\ & = \gcd_d\{\bar{A}_{ij} : 1 \le i \le m, 1 \le j \le n\} \\ & \ge 1/(16 \log \log d). \end{aligned}$$

Proof Let p be a prime dividing d, $e = \min\{\text{ord}_p(\bar{A}_{ij}) : 1 \le i \le m, 1 \le j \le n\}$ and k, l such that $\text{ord}_p(\bar{A}_{kl}) = e$ (where $\text{ord}_p(a) = \max\{i : p^i \mid a\}$ for $a \in \mathbb{Z}_d$). Let $x_2, \ldots, x_n$ be indeterminates and

$$f_k = \bar{A}_{k1} + \sum_{2 \le j \le n} x_j \bar{A}_{kj} \in \mathbb{Z}_d[x_2, \ldots, x_n].$$

Then $w_k = f_k(v_2, \ldots, v_n)$ and we can write $f_k = p^e g_k$, where $g_k \in \mathbb{Z}_d[x_2, \ldots, x_n] \setminus \{0\}$ and $g_k \not\equiv 0 \bmod p$. If each x_i is assigned a random v_i from $\mathbb{Z}_d$ for $2 \le i \le n$ then

$$\begin{aligned} \text{Prob}\{\text{ord}_p(w_k) = e\} &= \text{Prob}\{g_k(v_2, \ldots, v_n) \not\equiv 0 \bmod p\} \\ &\ge 1 - 1/p \end{aligned}$$

by Corollary 1 of Schwartz (1980). Thus

$$\begin{aligned} \text{Prob}\{\text{ord}_p(\, & \gcd_d(w_1, \ldots, w_m)) \\ & = \text{ord}_p(\gcd_d\{\bar{A}_{ij} : {\textstyle \substack{1 \le i \le m \\ 1 \le j \le n}}\})\} \\ & \ge 1 - 1/p. \end{aligned}$$

Since we choose $v_2, \ldots, v_n$ randomly and uniformly from $\mathbb{Z}_d$, the above probability is independent for each prime p dividing d. Thus,

$$\begin{aligned} \text{Prob}\Big\{ & \gcd_d(w_1, \ldots, w_m) = \gcd_d\{\bar{A}_{ij} : {\textstyle \substack{1 \le i \le m \\ 1 \le j \le n}}\}\Big\} \\ & = \prod_{p \mid d} \text{Prob}\left\{ \begin{array}{c} \text{ord}_p(\gcd_d(w_1, \ldots, w_m)) \\ = \text{ord}_p(\gcd_d\{\bar{A}_{ij} : {\textstyle \substack{1 \le i \le m \\ 1 \le j \le n}}\}) \end{array} \right\} \\ & \ge \prod_{p \mid d} (1 - 1/p) = \phi(d)/d, \end{aligned}$$

where $\phi(d)$ is Euler's totient function. By Rosser & Schoenfeld (1962), (3.41) and (3.42), $\phi(d)/d > 1/(1.8 \log\log d + 2.5036/\log\log d) > 1/(16 \log\log d)$ for all $d \geq 2$. □

We now present a Las Vegas algorithm for the Smith normal form algorithm of an $\bar{A} \in \mathbb{Z}_d^{m \times n}$

Algorithm: `ModSmithForm`

Input: $\bar{A} \in \mathbb{Z}_d^{m \times n}$;

Output: the non-zero invariant factors (in $\mathbb{Z}_d$) of $\bar{A}$;

(1) If $\bar{A}$ is the zero matrix then quit;

(2) If $m = 1$ then output $\gcd_d(\bar{A}_{11}, \ldots, \bar{A}_{1n}) \in \mathbb{Z}_d$ and quit;

(3) Choose random $v_2, \ldots, v_n \in \mathbb{Z}_d$. If $\bar{A} = [\bar{C}_1 | \cdots | \bar{C}_n]$, where $\bar{C}_i \in \mathbb{Z}_d^{m \times 1}$ is the ith column of $\bar{A}$, let $\bar{C}_1' := \bar{C}_1 + \sum_{2 \leq i \leq n} v_i \bar{C}_i \in \mathbb{Z}_d^{m \times 1}$ and $\bar{A}' := [\bar{C}_1' | \bar{C}_2 | \cdots | \bar{C}_n]$; if the first column of $\bar{A}'$ is all zeros, repeat (3);

(4) Find $\bar{P} \in \mathbb{Z}_d^{m \times m}$ with $\det \bar{P} = \pm 1$ such that $\bar{B} := \bar{P}\bar{A}'$ has $\bar{B}_{i1} = 0$ for $2 \leq i \leq m$;

(5) If $\bar{B}_{11} \mid \bar{B}_{ij}$ for $1 \leq i \leq m$, $1 \leq j \leq n$ then output $\bar{B}_{11}$ and call `ModSmithForm` on the submatrix $\bar{B}' \in \mathbb{Z}_d^{(m-1) \times (n-1)}$ of $\bar{B}$ formed by deleting the first row and first column, otherwise goto (3);

Theorem 1.3 *The algorithm* `ModSmithForm` *works correctly as specified. It requires an expected* $O(mn\bar{r} \log\log d)$ *operations in* $\mathbb{Z}_d$, *where* $\bar{r} = \operatorname{rank}(\bar{A})$, *the number of non-zero invariant factors of* $\bar{A}$ *in* $\mathbb{Z}_d$.

Proof In step (3) the transformation from $\bar{A}$ to $\bar{A}'$ is unimodular so the Smith form of $\bar{A}$ equals that of $\bar{A}'$. Since $\bar{B}_{11} = \gcd_d(\bar{A}'_{11}, \ldots, \bar{A}'_{m1}) \in \mathbb{Z}_d$, by Lemma 1.2 $\bar{B}_{11} \mid \bar{B}_{ij}$ in $\mathbb{Z}_d$ for $1 \leq i \leq m$, $1 \leq j \leq n$ with probability at least $1/16 \log\log d$. Thus, with this probability $\bar{B}_{11}$ is the first invariant factor of $\bar{A}$. If this is the case then $\bar{B}$ is column equivalent to $\begin{pmatrix} \bar{B}_{11} & 0 \\ 0 & \bar{B}' \end{pmatrix}$, and the remaining invariant factors of $\bar{A}$ are exactly the invariant factors of $\bar{B}'$. These are found by the recursion in step (5).

By Lemma 1.2 we execute steps (3)-(5) an expected number of $16 \log\log d$ times in each recursive call. These steps require $O(nm)$ operations in $\mathbb{Z}_d$ by Fact 1.1 and there are at most $\bar{r}$ recursive calls. □

Theorem 1.4 *Given an* $A \in \mathbb{Z}^{m \times n}$ *with* $m \leq n$, *we can compute the Smith normal form* S *of* A *with an expected* $O(m^2 n \cdot \mathrm{M}(m(\log \|A\| + \log m)))$ *bit operations. The algorithm is probabilistic of the Las Vegas type.*

Proof Compute $r = \operatorname{rank}(A)$ and a multiple d of the non-zero invariant factors of A. This can be done with

$$O(m^2 n \,\mathrm{M}(m(\log \|A\| + \log m)))$$

bit operations by Proposition 2.3 of Hafner & McCurley (1991). Next compute the Smith normal form

$$S = \operatorname{diag}(\bar{s}_1, \ldots, \bar{s}_r, 0, \ldots, 0) \in \mathbb{Z}_d^{m \times n}$$

of $\bar{A} = A \bmod d$. This requires $O(m^2 n)$ operations in $\mathbb{Z}_d$ by Theorem 1.3. Note that it may be that $\bar{r} = \operatorname{rank}(\bar{A}) < r$ in which case $\bar{s}_i = 0$ for some i, and $s_i = d$. This only happens if $d = 1$, in which case $\bar{r} = 0$ and $S = \operatorname{diag}(1, \ldots, 1, 0, \ldots, 0) \in \mathbb{Z}^{m \times n}$ (with r ones on the diagonal), or if $d > 1$, $S = \operatorname{diag}(1, \ldots, 1, d, 0, \ldots, 0)$ and $\bar{r} = r - 1$. The algorithm works correctly in both these instances.

For $1 \leq i \leq r$, let $d_i = \gcd(\bar{s}_1 \cdots \bar{s}_i, d) \in \mathbb{Z}$ and $s_1 = d_1$, $s_i = d_i/d_{i-1}$ for $2 \leq i \leq r$. The output is then $S = \operatorname{diag}(s_1, \ldots, s_r, 0, \ldots, 0) \in \mathbb{Z}^{m \times n}$. □

2 A faster, space efficient Smith form algorithm

In this section we present our new, faster and more space efficient Monte Carlo algorithm for the Smith normal form of an $A \in \mathbb{Z}^{m \times n}$ of rank r. The algorithm is related to that of Kaltofen *et al.* (1987) for computing Smith normal forms of matrices of polynomials over a field K. They show that for $A \in \mathsf{K}[x]^{n \times n}$, to compute the kth determinantal divisors d_k you need only consider the GCD of the leading minors of two random perturbations of A, and not of the $\binom{n}{k}^2$ minors of A indicated by the definition. Recall the *leading* $k \times k$ *minor* of any $B \in \mathbb{Z}^{m \times n}$ is $B\binom{1 \ldots k}{1 \ldots k}$. For randomly chosen $R, T, U, V \in \mathsf{K}^{n \times n}$ they prove that d_k equals the GCD of the leading $k \times k$ minors of RAT and UAV for *all* k $(1 \leq k \leq n)$ with high probability. They obtain a probabilistic algorithm in the parallel complexity class RNC^2 for the Smith normal form of a polynomial matrix.

Our new algorithm for the Smith normal form is based on a new characterization in Subsection 2.1 of the determinental divisors of A as the contents of some $f_1, \ldots, f_r \in \mathbb{Z}[x_1, \ldots, x_s]$, where $s = r(n + m)$ (recall that the content $\operatorname{cont}(f_i) \in \mathbb{Z}$ of f_i is the GCD of all the coefficients of f_i). While we cannot write down these $f_1, \ldots, f_r$ efficiently, we can evaluate them quickly, i.e., we provide a black box which computes $f_1(\vec{a}), \ldots, f_r(\vec{a}) \in \mathsf{R}$ for any $\vec{a} \in \mathsf{R}^s$, where R is a small extension ring of $\mathbb{Z}$. In Subsection 2.2 we show how to find the contents of a list of polynomials given by a black box, using only a small number of evaluations on "small" input points. The black box we need for the Smith normal form computation finds all the leading minors of a $B \in \mathsf{R}^{r \times r}$. In Subsection 2.3 we show how find these very quickly. Finally, in Subsection 2.4 we tie these results together to obtain a fast, space efficient, Monte Carlo algorithm for the Smith normal form of an integer matrix.

2.1 A new characterization of the determinental divisors

The following theorem gives a different and very useful characterization of the determinental divisors of a matrix as the content of the leading minors of a matrix of polynomials.

Theorem 2.1 *Let* $A \in \mathbb{Z}^{m \times n}$ *have rank* r. *Let* $X = (X_{ij})$ *be an* $r \times m$ *matrix and* $Y = (Y_{lk})$ *an* $n \times r$ *matrix of algebraically independent indeterminates in* $\Lambda = \{X_{ij}, Y_{lk} : 1 \leq i, k \leq r; 1 \leq j \leq m; 1 \leq l \leq n\}$. *Then the content of the* k*th leading minor* $f_k \in \mathbb{Z}[\Lambda]$ *of* $B = XAY \in \mathbb{Z}[\Lambda]^{r \times r}$ *equals the* k*th determinental divisor* $d_k \in \mathbb{Z}$ *of* A, *for* $1 \leq k \leq r$.

Proof Using the Binet-Cauchy formula (Gantmacher 1990, p. 9) we find

$$f_k = C\binom{1 \ldots k}{1 \ldots k} = \sum_{\substack{(b_1, \ldots, b_k) \in C_k^m \\ (c_1, \ldots, c_k) \in C_k^n}} X\binom{1 \ldots k}{b_1 \ldots b_k} A\binom{b_1 \ldots b_k}{c_1 \ldots c_k} Y\binom{c_1 \ldots c_k}{1 \ldots k}.$$

for $1 \leq k \leq r$. The kth determinental divisor d_k of A by definition divides $A\binom{b_1 \ldots b_k}{c_1 \ldots c_k}$ for all $(b_1, \ldots, b_k) \in C_k^m$ and $(c_1, \ldots, c_k) \in C_k^n$, so $d_k \mid \operatorname{cont}(f_k)$.

Now we show $\text{cont}(f_k) \mid d_k$. Assume to the contrary that there exists a prime p and integer e such that $p^e \mid \text{cont}(f_k)$ but $p^e \nmid d_k$. Thus, there exists an $k \times k$ minor $\lambda \in \mathbb{Z}$ of A such that $p^e \nmid \lambda$. Since $k \leq r$, there exists a $P \in \mathbb{Z}^{r \times m}$ and $Q \in \mathbb{Z}^{n \times r}$ such that λ is the leading minor of $PAQ \in \mathbb{Z}^{r \times r}$, so $f_k(P, Q) = \lambda$. Therefore λ is divisible by the content of f_k, whence $p^e \mid \lambda$, a contradiction. □

We do not compute the leading minors of $XAY \in \mathbb{Z}[\Lambda]^{r \times r}$ explicitely, but instead compute them by a *black-box* subroutine. That is, we will assign $X := P \in \mathsf{R}^{r \times m}$ and $Y := Q \in \mathsf{R}^{n \times r}$ for some randomly chosen P and Q over an extension ring R of $\mathbb{Z}$. We then find the leading minors of $PAQ \in \mathsf{R}^{r \times r}$. If $f_k \in \mathbb{Z}[\Lambda]$ is the leading $k \times k$ minor of XAY, then the leading $k \times k$ minor of PAQ is $f_k(P, Q) \in \mathsf{R}$. The algorithm `FindContent` in the next subsection employs this black box subroutine to find the contents of $f_1, \ldots, f_r$.

2.2 Finding the content of black-box polynomials

Our goal in this subsection is to find the contents of $f_1, \ldots, f_r \in \mathbb{Z}[x_1, \ldots x_s] \setminus \{0\}$, each with degree at most ν. 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 for an extension ring R of $\mathbb{Z}$ and $a_1, \ldots, a_s \in \mathsf{R}$ can compute

$$(f_1(a_1, \ldots, a_s), \ldots, f_r(a_1, \ldots, a_s)) \in \mathsf{R}^r$$

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.

Informally, the idea is as follows. We maintain a vector $(c_1, \ldots, c_r) \in \mathbb{Z}^r$ which contains an "approximation" to $(\text{cont}(f_1), \ldots, \text{cont}(f_r))$. 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 $d_i = \text{cont}(f_i) \in \mathbb{Z}$ 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 $(b_1, \ldots, b_r) := f(\vec{a})$. Now 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 $p \mid d_i$, $\text{ord}_p(c_i^{(j-1)}) \geq \text{ord}_p(c_i^{(j)}) \geq \text{ord}_p(\text{cont}(f_i))$ 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 $\text{ord}_p(d_i) = \text{ord}_p(c_i^{(j)})$ is quite high. Iterating this processes allows us to make the probability of error arbitrarily small.

Unfortunately, this method does not necessarily work for small primes p, and the order of such a p in c_i may converge slowly, or even not at all, to the order of p in d_i. However, by working in a specially constructed extension ring R of $\mathbb{Z}$ we are able to prove sufficiently fast convergence. 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 $\mathsf{R} = \mathbb{Z}[x]/(\Gamma)$, where $\Gamma = x^2 + x + 1 \in \mathbb{Z}[x]$, we can choose $a = (x \bmod \Gamma)$, so $f(a) = (4x - 2 \bmod \Gamma) \in \mathsf{R}$. GCD's are not 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 $-4x - 2$ is 2, as is the content of f. We prove that for any polynomial f, the corresponding sequence of GCD's of contents of evaluations of f in $\mathsf{R} = \mathbb{Z}[x]/(\Gamma)$ converges quickly on the content of f with high probability, for an appropriately selected Γ.

In our applications of this algorithm, the black box we are given may not work on all inputs. We allow for the existence of a $\chi \in \mathbb{Z}[x_1, \ldots, x_s]$ of degree $O(\nu^2)$ such that for $\vec{a} \in \mathsf{R}^s$, the black box is only guaranteed to find $(f_1(\vec{a}), \ldots, f_r(\vec{a}))$ when $\chi(\vec{a}) \in \overline{\mathsf{R}}^*$. Here $\overline{\mathsf{R}} = \mathbb{Q} \otimes \mathsf{R}$ and $\overline{\mathsf{R}}^*$ is the set of units in $\overline{\mathsf{R}}$. When $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$, the black box may report "failure" but cannot produce an incorrect answer.

A variant of Schwartz's Lemma

To prove that the algorithm sketched above to find the contents of black-box polynomials converges we will require a variant of Corollary 1 of Schwartz (1980), "Schwartz's Lemma". This bounds from above the probability that a random point, with coordinates chosen randomly and uniformly from a finite set $\mathcal{V}$, is a non-zero of a polynomial. We require a version of this in which coordinates are not chosen uniformly from $\mathcal{V}$, but where we only have an upper bound on the probability of choosing any one element of $\mathcal{V}$.

Lemma 2.2 *Assume $f \in \mathsf{D}[x_1, \ldots, x_k]$ is non-zero, D an integral domain, and $\mathcal{V}$ a finite subset of D. Suppose elements $a_1, \ldots, a_k$ are randomly chosen from $\mathcal{V}$ such that each a_i is assigned any one element of $\mathcal{V}$ with probability at most ϱ. Then* $\text{Prob}\{f(a_1, \ldots, a_k) = 0 : a_1, \ldots, a_k \in \mathcal{V}\} \leq \varrho \deg f$.

Proof Follows almost identically to Corollary 1 of Schwartz (1980). □

Building a rough extension of $\mathbb{Z}$

Unfortunately Lemma 2.2 above, like Schwartz's (1980) Corollary 1, is of little use when the size of $\mathcal{V}$ is less than $\deg f$. In particular, when $\mathsf{D} = \mathbb{Z}_p$, $\#\mathcal{V} \leq p$ and $\tau \geq 1/p$. When $\deg f \geq p$, Lemma 2.2 is trivial. Our solution will be to constuct a *rough* extension ring R of $\mathbb{Z}$, one such that $\mathsf{R} \bmod p$ contains a large finite field for each of selected set of primes p. We show that the degree of R over $\mathbb{Z}$ can be kept surprisingly small.

Let $\eta \geq 1$ and $p_1, \ldots, p_\kappa$ be primes with $2 \leq p_1, \ldots, p_\kappa \leq \tau$ for some $\tau \geq 2$. We next show how to construct a monic, squarefree $\Gamma \in \mathbb{Z}[x]$ of small height and degree $\gamma = O(\eta \log \kappa)$ such that for each i $(1 \leq i \leq \kappa)$, $\Gamma \bmod p_i$ has a factor of degree greater than η in $\mathbb{Z}_p[x]$ (i.e., the factorization of Γ is "rough" modulo each prime p_i). The *height* $\|\Gamma\|$ of $\Gamma = c_0 + c_1 x + \cdots + c_\gamma x^\gamma \in \mathbb{Z}[x]$ is defined as $\max_{0 \leq i \leq \gamma} |c_i|$. Notice that the degree of Γ is *logarithmic* in the number of primes. This will be important in our application, where κ may be relatively large.

For any fixed prime p and $n \in \mathbb{N}$, define

$$\mathcal{M}_p(n) = \{g \in \mathbb{Z}_p[x] : g \text{ monic and } \deg g = n\},$$

$$\mathcal{I}_p(n) = \left\{ g \in \mathbb{Z}_p[x] : g \text{ monic and irreducible and } \deg g = n \right\}.$$

We first give a lower bound on

$$R_p(\eta) = \#\{f \in \mathcal{M}_p(2\eta) : \exists k > \eta,\ g \in \mathcal{I}_p(k) \text{ such that } g \mid f\},$$

the number of monic polynomials in $\mathbb{Z}_p[x]$ of degree 2η which have an irreducible factor in $\mathbb{Z}_p[x]$ of degree greater than η.

Lemma 2.3 *For a prime p and integer $\eta \geq 1$ we have $R_p(\eta) > p^{2\eta}/3$.*

Proof Since any $f \in \mathcal{M}_p(2\eta)$ can have at most one irreducible factor with degree greater than η, we obtain the formula

$$R_p(\eta) = \sum_{\eta < i \le 2\eta} p^{2\eta - i} N_p(i)$$

where $N_p(i) = \#\mathcal{I}_p(i)$. Using the lower bound on $N_p(i)$ given by Lidl & Niederreiter(1983, Exercise 3.27) and Euler's summation formula, we get

$$R_p(\eta) > p^{2\eta}\left(\sum_{\eta < i \le 2\eta} \frac{1}{i} - \frac{p}{p-1}\sum_{\eta < i \le 2\eta} \frac{1}{ip^{i/2}}\right) > p^{2\eta}\cdot s(p,\eta),$$

where

$$s(p,\eta) = \log 2 - \frac{1}{2\eta} - \frac{p}{p-1}\cdot\frac{1}{\eta}\cdot\frac{\sqrt{p}}{\sqrt{p}-1}\cdot\frac{1}{p^{\eta/2+1/2}}.$$

The function $s(p,\eta)$ is strictly increasing in both p and η and is greater than $1/3$ with the exceptions $p = 2$, $2 \le \eta \le 4$, and $p = 3$, $\eta = 2$. Excepting these cases, $R_p(\eta) > s(p,\eta) > p^{2\eta}/3$. It is easily checked, using the exact formula $N_p(i) = (1/i)\cdot\sum_{d|i}\mu(d)p^{i/d}$ for $i \ge 1$ (where μ is the Mobius function), that indeed $R_p(\eta) > p^\eta/3$ in the exceptional cases as well. □

Let $\mathcal{V} = \{g \in \mathbb{Z}[x] : g \text{ monic, } \deg g = 2\eta, \text{ and } \|g\| \le 2\eta\tau\}$. If we choose f randomly and uniformly from $\mathcal{V}$, f falls into a particular residue class in $\mathcal{M}_p(2\eta)$ with probability at least $(\lfloor(4\eta\tau+1)/p\rfloor/(4\eta\tau+1))^{2\eta} \ge (1/p - 1/(4\eta\tau+1))^{2\eta}$. Thus by Lemma 2.3 the probability that $f \bmod p$ has an irreducible factor modulo p of degree greater than η is at least $(1/p - 1/(4\eta\tau+1))^{2\eta}\cdot p^{2\eta}/3 > 1/6$.

The following algorithm constructs a $\Gamma \in \mathbb{Z}[x]$ as required.

Algorithm: `BuildRoughExtension`

Input: $\eta \in \mathbb{Z}$ and primes $2 \le p_1, \ldots, p_\kappa \le \tau$;

Output: a squarefree, monic $\Gamma \in \mathbb{Z}[x]$ such that for each i $(1 \le i \le \kappa)$, $\Gamma \bmod p_i$ has an irreducible factor in $\mathbb{Z}_{p_i}[x]$ of degree greater than η.

(1) Repeat
(2) Let $\mathcal{P} := \{1, \ldots, \kappa\}$; $H := \{\}$;
(3) For $i := 1$ to $4 + 6\log\kappa$ while $\mathcal{P} \ne \{\}$ do
(4) Choose a random $h_i \in \mathcal{V}$;
(5) For $j \in \mathcal{P}$ do
(6) If $h_i \bmod p_j \in \mathbb{Z}_{p_j}[x]$ has an irreducible factor modulo p_j of degree greater than η Then $\mathcal{P} := \mathcal{P} \setminus \{j\}$; $H := H \cup \{h_i\}$;
End For;
End For;
Until $\mathcal{P} = \{\}$;
Return $\Gamma = \prod_{h\in H} h \in \mathbb{Z}[x]$;

Theorem 2.4 *The algorithm* `BuildRoughExtension` *always produces the correct results as described and requires an expected number of* $O((\eta^3 + \eta^2\log\tau)\cdot\kappa\log^2\tau\log\kappa)$ *bit operations. The output* $\Gamma \in \mathbb{Z}[x]$ *has degree at most* $2\eta(4 + 6\log\kappa) = O(\eta\log\kappa)$ *and* $\|\Gamma\| \le (4\eta^2\tau)^{\log\kappa}$.

Proof We first examine the probability that the algorithm successfully finds a Γ in an iteration of the outer loop, or equivalently, finds an $H \subseteq \mathcal{V}$ such that for each i, there exists an $h \in H$ such that $h \bmod p_i$ has an irreducible factor in $\mathbb{Z}_p[x]$ of degree greater than η. For fixed j, the probability that $h_i \bmod p_j$ has no factor in $\mathcal{I}_{p_j}(r)$ for some $r > \eta$, for *all* $1 \le i \le l$ is less than $(5/6)^l$ by Lemma 2.3. The probability this is true for all j is less than $\kappa\cdot(5/6)^l < 1/2$ by our choice of $l = 4 + 6\log\kappa \ge -\log(2\kappa)/\log(5/6)$.

For each random choice of $h_i \in \mathcal{V}$ the inner loop of steps (5)-(6) can be accomplished with an expected number of $O((\eta^3 + \eta^2\log\tau)\log^2\tau\cdot\kappa)$ bit operations using Berlekamp's (1970) factoring algorithm, and this loop is executed $4 + 6\log\kappa$ times per iterations of the outer loop.

For any $h_1, h_2 \in \mathbb{Z}[x]$, $\|h_1 h_2\| \le \min(\deg h_1, \deg h_2)\cdot\|h_1\|\|h_2\|$. Since Γ is the product of $O(\log\kappa)$ polynomials of degree 2η and height at most $2\eta\tau$, it follows that $\|\Gamma\| \le (4\eta^2\tau)^{\log\kappa}$. □

We define $\mathsf{R} = \mathbb{Z}[x]/(\Gamma)$, an extension ring of R where $\Gamma \in \mathbb{Z}[x]$ is monic of degree γ and is constructed using `BuildRoughExtension` on some η and primes $p_1, \ldots, p_\kappa$. The ring $\mathsf{R}_{p_i} = \mathsf{R} \bmod p_i$ contains a copy of $\mathrm{GF}(p_i^{\xi_i})$ for some $\xi_i > \eta$ for $1 \le i \le \kappa$. We represent an $a \in \mathsf{R}$ by its *least degree residue* $\hat{a} \in \mathbb{Z}[x]$ with $a \equiv \hat{a} \bmod \Gamma$ and $\deg\hat{a} < \gamma$. The notion of height can be extended to R by $\|a\| = \|\hat{a}\|$.

The probability of correctly finding the content

We return the the problem of obtaining the content $d_i \in \mathbb{Z}$ of $f_i \in \mathbb{Z}[x_1, \ldots, x_s]$ where $\deg f_i \le \nu$, for $1 \le i \le r$. 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 $\Gamma \in \mathbb{Z}[x]$ be monic of degree $\gamma > 1$ and $\mathsf{R} = \mathbb{Z}[x]/(\Gamma)$. Let $\beta \ge 5$ be an integer with $2\beta \ge \nu$, and let $L = \{-\beta, \ldots, \beta\} \subseteq \mathbb{Z}$. Let $\eta \le \deg\Gamma$ be an integer and $\mathcal{W} = \{h \bmod \Gamma : h \in L[x], \deg h < \eta\} \subseteq \mathsf{R}$. We will further specify our choices of Γ, γ, β and η 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 $\bar{b}_i \in \mathbb{Z}[x]$, $\deg\bar{b}_i < \deg\Gamma$ and $\bar{b}_i \equiv b_i \bmod \Gamma$. We have $\overline{\mathrm{cont}}(b_i) = d_i\,\overline{\mathrm{cont}}(g_i(\vec{a}))$ so $\mathrm{ord}_p(d_i) \le \mathrm{ord}_p(\overline{\mathrm{cont}}(b_i))$, with equality when $g_i(\vec{a}) \not\equiv 0 \bmod p$.

In the next two lemmas we examine the probability that, 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`.

Lemma 2.5 *Let* $p > 2\beta$ *be prime and* $g \in \mathbb{Z}[x_1, \ldots, x_s]$ *with* $\deg g \le \nu$ *and* $\mathit{cont}(g) = 1$ *($2\beta \ge \nu$ as above). Suppose* $\vec{a}$ *is chosen randomly and uniformly from* $\mathcal{W}^s$. *Then* $\mathrm{Prob}\{g(\vec{a}) \equiv 0 \bmod p\} \le \nu/(2\beta)$.

Proof Assume $\vec{a} = (\bar{a}_1, \ldots, \bar{a}_s) \in \mathcal{W}^s$ where $\bar{a}_i \equiv a_i \bmod \Gamma$ and $a_i = \sum_{0 \le j < \eta} a_{ij}x^j \in \mathbb{Z}[x]$ for $a_{ij} \in L$. Then $g(\vec{a}) \equiv 0 \bmod p \iff g(a_1, \ldots, a_s) \equiv 0 \bmod (\Gamma, p)$. Assume for now that the a_{ij}'s are independent indeterminates over $\mathbb{Q}$. Let $\Lambda = \{a_{ij} : 1 \le i \le s, 0 \le j < \eta\}$, and define $\hat{g} = g(a_1, \ldots, a_s) \in \mathbb{Z}[\Lambda][x]$. Consider the division of $\hat{g}$ by Γ in $\mathbb{Q}(\Lambda)[x]$ to obtain remainder $\rho = \sum_{0 \le k < \gamma}\rho_k(\Lambda)x^k \in \mathbb{Z}[\Lambda][x]$, where $\rho_k \in \mathbb{Z}[\Lambda]$ has degree at most ν for $0 \le k < \gamma$. Now $g \not\equiv 0 \bmod p$ and $p > \deg g$, so there exists a $\vec{b} \in \mathbb{Z}^s$ such that $g(\vec{b}) \not\equiv 0 \bmod p$. Since $g(\vec{b}) \in \mathbb{Z}$, $\rho_0 \not\equiv 0 \bmod p$.

Again assuming $\vec{a}$ is randomly selected from $\mathcal{W}^s$, a necessary condition for $g(\vec{a}) \equiv 0 \bmod p$ is that $\rho_0(\vec{a}) \equiv 0 \bmod p$, whence Prob $\{g(\vec{a}) \equiv 0 \bmod p\} \leq$ Prob $\{\rho_0(\vec{a}) \equiv 0 \bmod p\} \leq \nu/(2\beta+1)$ by Corollary 1 of Schwartz (1980). □

Lemma 2.6 *Let $\beta \geq 5$ and $p \leq 2\beta$ prime, and assume $\Gamma \bmod p$ has an irreducible factor $\Upsilon \in \mathbb{Z}_p[x]$ of degree greater than η. Let $g \in \mathbb{Z}[x_1, \ldots, x_s]$ have degree $\leq \nu$ with cont$(g) = 1$. For a randomly and uniformly chosen $\vec{a} \in \mathcal{W}^s$,*

$$\text{Prob}\{g(\vec{a}) \equiv 0 \bmod p\} \leq \nu \cdot (3/5)^\eta.$$

Proof A randomly chosen $a \in \mathcal{W}$ lies in a particular residue class of $\mathcal{W}$ mod p with probability at most

$$\left(\left\lceil \frac{2\beta+1}{p} \right\rceil \cdot \frac{1}{2\beta+1}\right)^\eta \leq \left(\frac{1}{p} + \frac{1}{2\beta+1}\right)^\eta \leq (3/5)^\eta$$

for $\beta \geq 5$ and $p \leq 2\beta$. Since $\deg \Upsilon \geq \eta$ (and each $a \in \mathcal{W}$ has $\deg a < \eta$), the probability that a is in a particular residue class of $\mathcal{W}$ mod (p, Υ) is also at most $(3/5)^\eta$. Applying Lemma 2.2, Prob$\{g(\vec{a}) \equiv 0 \bmod (p, \Upsilon)\} \leq \nu \cdot (3/5)^\eta$, whence Prob$\{g(\vec{a}) \equiv 0 \bmod p\} \leq \nu \cdot (3/5)^\eta$. □

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

Lemma 2.7 *Let $\chi \in \mathbb{Z}[x_1, \ldots, x_s] \setminus \{0\}$. For a randomly and uniformly chosen $\vec{a} \in \mathcal{W}^s$, Prob$\{\chi(\vec{a}) \notin \overline{\mathsf{R}}^*\} \leq \deg(\chi) \cdot \deg(\Gamma)/(2\beta)$.*

Proof Assume $\vec{a} = (\bar{a}_1, \ldots, \bar{a}_s) \in \mathcal{W}^s$ where $\bar{a}_i = a_i \bmod \Gamma$ and $a_i = \sum_{0 \leq j < \eta} a_{ij} x^j \in \mathbb{Z}[x]$ for $1 \leq i \leq s$. Then $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ if and only if $\gcd(\chi(a_1, \ldots, a_s), \Gamma) \neq 1$ in $\mathbb{Q}[x]$, which is true if and only if the resultant $\text{res}(\chi(a_1, \ldots, a_s), \Gamma)$ is zero.

Now assume the a_{ij}'s above are algebraically independent indeterminates over $\mathbb{Q}$ and let $\Lambda = \{a_{ij} : 1 \leq i \leq s, 0 \leq j < \eta\}$. Then $a_i = \sum_{0 \leq j < \eta} a_{ij} x^j \in \mathbb{Z}[\Lambda][x]$ and $\text{res}(\chi(a_1, \ldots, a_s), \Gamma) \in \mathbb{Z}[\Lambda]$ has degree at most $\deg(\chi) \cdot \deg(\Gamma)$. If the a_{ij}'s are assigned uniformly and randomly values from L, then by Schwartz's (1980) Corollary 1, the probability that $\text{res}(\chi(a_1, \ldots, a_s), \Gamma) = 0$, and hence the probability that $\chi(a_1, \ldots, a_s) \notin \overline{\mathsf{R}}^*$, is at most $\deg(\chi) \cdot \deg(\Gamma)/(2\beta+1)$. □

Algorithm: `FindContent`

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

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

(1) Let $\beta := \max(r\nu + \deg \chi, \nu \cdot \deg(\chi) \log(\deg(\chi)), \nu^2, 5)$;

(2) Choose a random $\vec{a} \in \{-\beta, \ldots, \beta\}^s$;
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 $\leq 2\beta$ which divide $\text{lcm}(c_1^{(0)}, \ldots, c_r^{(0)})$;

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

(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 $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$, this choice of $\vec{a}$ may be ignored, and execution continued with next i at step (5);

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

(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 2.8 *The algorithm* `FindContent` *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 $\mathsf{R} = \mathbb{Z}[x]/(\Gamma)$, 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 $(r\nu)^{O(1)}$.*

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

Proof Step (2) finds a non-zero multiple c_i of d_i for $1 \leq i \leq r$. For a randomly chosen $\vec{a} \in \{-\beta, \ldots, \beta\}$

$$\begin{aligned} &\text{Prob}\{\ \chi(\vec{a}) \neq 0, f_i(\vec{a}) \neq 0 \text{ for all } 1 \leq i \leq r\} \\ &\quad = \text{Prob}\{(\chi \cdot f_1 \cdots f_r)(\vec{a}) \neq 0\} \\ &\quad \geq 1 - (r\nu + \deg \chi)/(2(r\nu + \deg \chi)) \geq 1/2 \end{aligned}$$

by Corollary 1 of Schwartz (1980). Thus we expect to evaluate the black box two times on points of height $O(\beta)$ in step (2). On completion of (2) we have

$$\begin{aligned} |c_i^{(0)}| = |f_i(\bar{a})| &\leq \|f_i\| \sum_{0 \leq j \leq \nu} \binom{j+s-1}{j} \beta^j \\ &= O(\|f_i\| \cdot \beta^\nu (\nu+s)^\nu), \end{aligned}$$

since there are at most $\binom{j+s-1}{j}$ monomials of degree j in f_i.

Assume for now we choose $\vec{a}$ randomly in step (6) from $\mathcal{W}$ and do not eliminate the cases when $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ in step (7). Fix an i between 1 and r. For a prime $p > 2\beta$ dividing d_i,

$$\begin{aligned} \text{Prob}\{\text{ord}_p(\bar{b}_i) \neq \text{ord}_p(d_i)\} &= \text{Prob}\{g_i(\vec{a}) \equiv 0 \bmod p\} \\ &\leq \nu/(2\beta) \leq \nu/(2\nu^2) \leq 1/\nu \end{aligned}$$

by Lemma 2.5. For a prime $p \leq 2\beta$ dividing d_i

$$\begin{aligned} &\text{Prob}\{\text{ord}_p(\bar{b}_i) \neq \text{ord}_p(d_i)\} = \text{Prob}\{g_i(\vec{a}) \equiv 0 \bmod p\} \\ &\leq \nu \cdot (3/5)^\eta \leq \nu \cdot (3/5)^{4\log \nu} \leq 1/\nu \end{aligned}$$

by Lemma 2.6. Thus, for any prime p, $\text{Prob}\{\text{ord}_p(\tilde{b}_i) \neq \text{ord}_p(d_i)\} \leq 1/\nu$.

The probability that $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$, and hence the probability that the black box might not work, is at most

$$\begin{aligned}\deg(\chi)\deg(\Gamma)/(2\beta) &= 4\deg(\chi)\cdot\log(\nu)\cdot(4+6\log(\beta))/\beta \\ &\leq 81\log^2(\nu)/\nu,\end{aligned}$$

using Lemma 2.7 and the fact that $\kappa \leq \beta$ for $\beta \geq 5$. Thus, for a fixed i and prime p dividing $c_i^{(0)}$, the probability that $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ *or* $\text{ord}_p(\tilde{b}_i) \neq \text{ord}_p(d_i)$ is at most $1/\nu + 81\log^2(\nu)/\nu \leq 82\log^2(\nu)/\nu$.

Let $\omega(c_i^{(0)})$ denote the number of distinct primes dividing $c_i^{(0)}$. The probability that after l iterations of the loop there exists an i and a prime p dividing $c_i^{(0)}$ such that for all l randomly chosen $\vec{a} \in \mathsf{R}^s$ we have $\chi(\vec{a}) \notin \overline{\mathsf{R}}^*$ or $\text{ord}_p(c_i^{(l)}) \neq \text{ord}_p(d_i)$ is at most

$$\begin{aligned}&\sum_{1\leq i\leq r}\omega(c_i^{(0)})\cdot(82\log^2(\nu)/\nu)^l \\ &\qquad\leq r\log_2\max|c_i^{(0)}|(82\log^2(\nu)/\nu)^l \leq \epsilon\end{aligned}$$

by our choice of l. Thus, the probability that the algorithm produces the correct answer is at least $1-\epsilon$, as required. The number of evaluations of the black box is $l = O((\log(1/\epsilon) + \log r + \log\log s + \log\log\max_i\|f_i\|)/\log\nu)$.

Note that $\deg\Gamma = O(\eta\log\kappa) = O(\log(\nu)\log(r))$ and $\log\|\Gamma\| = O((\log r + \log\nu + \log\deg\chi)^2)$ by Lemma 2.4. Arguments $(a_1,\ldots,a_s) \in \mathcal{W}^s$ to the black box satisfy $\|a_i\| \leq \beta = (r\nu\deg\chi)^{O(1)}$ for $1 \leq i \leq s$.

The cost (in addition to the cost of the black-box evaluations) is dominated by the loop in steps (5)-(9). For $1 \leq i \leq r$, $\log\|b_i\| = O(\log\|f_i\| + \nu\log(\nu+s) + \nu\log\beta + \nu\gamma\log\|\Gamma\|)$, so the cost of l iterations of steps (8)-(9) is $O^\sim(lr(\nu+\log\|f_i\|)^2)$ bit operations using standard arithmetic. At any time we store $O(r)$ elements of R with $O^\sim(\nu+\log\max_i\|f_i\|)$ bits each and s elements with $O(\log r + \log\nu)$ bits each. □

2.3 Finding the leading minors of a matrix over R

We now show how to find all the leading minors of a $B \in \mathsf{R}^{r\times r}$. Again $\mathsf{R} = \mathbb{Z}[x]/(\Gamma)$ is an extension ring of $\mathbb{Z}$, where $\Gamma \in \mathbb{Z}[x]$ is monic of degree γ, and $\overline{\mathsf{R}} = \mathbb{Q}[x]/(\Gamma)$. For a prime p, let $\mathsf{R}_p = \mathsf{R} \bmod p$. We attempt to determine the leading minors of B by finding matrices $L, U \in \mathsf{R}_p^{r\times r}$ such that L is lower triangular with ones on the diagonal, U is upper triangular, and $B \equiv LU \bmod p$ — an LU-decomposition of B mod p. If such a decomposition exists, the kth leading minor d_k of B satisfies $d_k \equiv \prod_{1\leq i\leq k} U_{ii} \bmod p$. We can recover each d_k by Chinese remaindering with sufficiently many primes.

In what follows, let E be any commutative ring.

Lemma 2.9 *Let $B \in \mathsf{E}^{r\times r}$ such that all leading minors are in E^*. Then there exists a lower triangular matrix $L \in \mathsf{E}^{r\times r}$ with ones on the diagonal, and an upper triangular matrix $U \in \mathsf{E}^{r\times r}$, such that $B = LU$.*

Proof The proof is essentially the same as for the existence of an LU-decomposition over $\mathbb{R}$ without pivoting (see Golub & Van Loan 1983, Section 4.2). □

Lemma 2.10 *Let $B \in \mathsf{E}^{r\times r}$. Then we can either find an upper triangular $U \in \mathsf{E}^{r\times r}$, and a lower triangular $L \in \mathsf{E}^{r\times r}$ with ones on its diagonal, such that $B = LU$, or report that one of the leading minors is not in E^*, using $O(r^\theta)$ operations in E.*

Proof This follows from an application of the asymptotically fast LUP-decomposition algorithm of Aho *et al.* (1974, section 6.4). This returns the factorization $B = LUP$, for L, U as above and P an $n \times n$ permutation matrix. When all leading minors of B lie in E^*, it is easily proven by induction on the number of rows in B that the returned P is an identity matrix. Simply run the LUP decomposition algorithm and if at any stage the returned $P \neq I$ report that one of the leading minors is not in E^*. Aho *et al.*'s (1974) algorithm requires $O(r^\theta)$ operations in E. □

We again assume that $B \in \mathsf{R}^{r\times r}$ and define $\|B\| = \max\{\|B_{ij}\| : 1 \leq i,j \leq r\}$. We require a simple upper bound for $\|\det B\|$ in terms of $\|B\|$, $\|\Gamma\|$, and $\gamma = \deg\Gamma$.

Lemma 2.11 *Let $\Gamma \in \mathbb{Z}[x]$ be monic of degree γ and $\mathsf{R} = \mathbb{Z}[x]/(\Gamma)$. For $B \in \mathsf{R}^{r\times r}$ we have $\|\det B\| < 2^{r\gamma}(r\gamma)^r \cdot \|B\|^r\|\Gamma\|^{r\gamma}$.*

Proof Let $\hat{B} \in \mathbb{Z}[x]^{r\times r}$ be such that $\deg\hat{B}_{ij} \leq \gamma - 1$ and $\hat{B} \equiv B \bmod \Gamma$. Then $\det\hat{B}$ is a sum of $r!$ polynomials of degree at most $r(\gamma-1)$, each of is a product of r elements of $\hat{B}$ and so has height at most $\gamma^{r-1}\|B\|^r$. (see Giesbrecht 1993, Theorem 1.5 for heights of products and division with remainder of integer polynomials). Doing division with remainder by Γ we find $\|\det B\| \leq 2^{r\gamma}(r\gamma)^r \cdot \|B\|^r\|\Gamma\|^{r\gamma}$. □

Theorem 2.12 *Let $B \in \mathsf{R}^{r\times r}$. We can construct an algorithm which either returns all the leading minors of B or reports that one of them is not a unit in R. The algorithm requires $O^\sim(r^{\theta+1}\cdot(\gamma^2\log\|B\| + \gamma^4\log\|\Gamma\|))$ bit operations using standard integer and polynomial arithmetic. The algorithm is probabilistic of the Las Vegas type and requires space for $O^\sim(r^2(\log\|B\| + \gamma\log\|\Gamma\|))$ bits.*

Proof Let $h \in \mathbb{Z}[x]$ with $h \equiv \det(B) \bmod \Gamma$ with $\deg h < \gamma$. By Lemma 2.11, $\|h\| < \|B\|^r\|\Gamma\|^{r\gamma}(r\gamma)^r 2^{r\gamma}$.

Assume for now that all the leading minors of B lie in $\overline{\mathsf{R}}^*$ so $B = LU$, where $L \in \overline{\mathsf{R}}^{r\times r}$ is lower triangular with ones on the diagonal and $U \in \overline{\mathsf{R}}^{r\times r}$ is upper triangular. The kth leading minor of B is then $m_k = \prod_{1\leq i\leq k} U_{ii} \in \mathsf{R}$. For a prime p, we can compute $m_1 \bmod p, \ldots, m_r \bmod p$ when each or these is in R_p^*. This is true if and only if $\text{res}(h,\Gamma) \not\equiv 0 \bmod p$. Applying Hadamard's bound to the Sylvester matrix of Γ and h, we find

$$|\text{res}(h,\Gamma)| \leq \gamma^\gamma\|h\|^\gamma\|\Gamma\|^\gamma \leq \xi := \gamma^\gamma\|B\|^{r\gamma}\|\Gamma\|^{r\gamma^2}(r\gamma)^{r\gamma}2^{r\gamma^2}.$$

Let $\mu = \|B\|^r\|\Gamma\|^{r\gamma}(r\gamma)^r 2^{r\gamma}$ — by Lemma 2.11, μ is greater than the height of any minor of B. Let z be such that the product of the z smallest primes is greater than $2\mu\xi$, and let $\mathcal{P}$ be the set containing the smallest z primes. It must be the case that the primes in $\mathcal{P}$ which do not divide $\text{res}(h,\Gamma)$ have product greater than twice the height of any leading minor of B. If the kth minor $m_k \in \mathsf{R}$ of B is not in $\overline{\mathsf{R}}^*$ then $(m_k \bmod p) \notin \mathsf{R}_p^*$ for any prime p.

We proceed by computing an LU-decomposition of B modulo each $p \in \mathcal{P}$. We either obtain the leading minors of B mod p in R mod p or report that one of the leading

minors of B mod p is a non-unit in R mod p. Let $\mathcal{Q} \subseteq \mathcal{P}$ be the primes p modulo which each minor of B mod p is a unit in R mod p – we can computed an LU-decomposition of B mod p. If $\prod_{p \in \mathcal{Q}} p \geq \mu$ then we can recover all the leading minors of B by Chinese remaindering, and we return these. Otherwise, we report that one of the leading minors of B is not a unit in $\overline{\mathsf{R}}$.

To analyse the cost of this algorithm we note

$$z = \pi\left(\min_y \{e^{\vartheta(y)} \geq \mu\xi\}\right) = O\left(\frac{r\log\|B\| + r\gamma^2\log\|\Gamma\| + r\gamma\log(r\gamma)}{\log\log\|B\| + \log\log\|\Gamma\| + \log r + \log\gamma}\right)$$

where $\vartheta(y)$ is the Chebyshev theta function (the log of the product of all primes less than y), and $\pi(y)$ is the prime number function (the number of primes less than y). It is well known that $\vartheta(y) = O(\log y)$ and $\pi(y) = y/\log(y)$. All primes in $\mathcal{P}$ have $l = O(\log\log\|B\| + \log\log\|\Gamma\| + \log r + \log\gamma)$ bits. For a prime $p \in \mathcal{P}$ we can do an operation in R mod p with $O(\gamma^2 l^2)$ bit operations, and we can find the leading minors of $(B \bmod p) \in (\mathsf{R} \bmod p)^{r\times r}$ with $O(r^\theta\gamma^2 l^2)$ bit operations. To do this for all z primes requires $O(zr^\theta\gamma^2 l^2)$ or $O^\sim(r^{\theta+1}\cdot(\gamma^2\log\|B\| + \gamma^4\log\|\Gamma\|))$ bit operations. The Chinese remaindering needed to recover the leading minors in R can also be done in this time. The space required for the output is dominant, and this is $O(r\log\mu)$ or $O^\sim(r^2(\log\|B\| + \gamma\log\|\Gamma\|))$ □

2.4 Monte Carlo computation of the Smith normal form

In this subsection we present our new Monte Carlo algorithm for computing the Smith normal form of an integer matrix. It is based on a reduction to finding the contents of a list of black box polynomials, as discussed above.

We require an asymptotically fast algorithm for finding the rank of an $A \in \mathbb{Z}^{m\times n}$.

Lemma 2.13 *Let $A \in \mathbb{Z}^{m\times n}$ with $m \leq n$. We can compute $r = \mathrm{rank}(A)$ with $O(m^\theta n\log\|A\|)$ bit operations using standard integer arithmetic.*

Proof We use a standard homomorphic imaging scheme. Compute a number z such that $b = \prod_{p\leq z}^{p \text{ prime}} p > m^{m/2}\|A\|^m$. By Hadamard's bound every minor of A is less than b. Suppose $c \in \mathbb{Z}$ is an $r \times r$ minor of A. For each prime $p \leq z = O(m(\log m + \log\|A\|))$ find the rank of A mod p. Ibarra *et al.* (1982) show this can be accomplished with $O(m^{\theta-1}n\log^2 p)$ bit operations. At least one prime $p \leq z$ does not divide c, and $\mathrm{rank}(A \bmod p)$ is maximal and equal to $r = \mathrm{rank}(A)$. □

Theorem 2.14 *Let $A \in \mathbb{Z}^{m\times n}$ with $m \leq n$, and ϵ a positive constant. We can construct a Monte Carlo type probabilistic algorithm to compute the Smith normal form of A which requires an expected number of $O^\sim((m^\theta n\log\|A\| + m^3\log^2\|A\|)\cdot\log(1/\epsilon))$ bit operations using standard integer and polynomial arithmetic. It returns the correct answer with probability at least $1-\epsilon$, and requires $O^\sim(nm\log\|A\|)$ bits of storage.*

Proof By Theorem 2.1 we need only find the content d_k of the kth leading minor $f_k \in \mathbb{Z}[\Lambda]$ of $XAY \in \mathbb{Z}[\Lambda]^{r\times r}$ for $1 \leq k \leq r$, where Λ, $X \in \mathbb{Z}^{r\times m}$, and $Y \in \mathbb{Z}[\Lambda]^{n\times r}$ are as in Theorem 2.1. By Theorem 2.8, the algorithm `FindContent` does just this. The height of f_i is $O(\|A\|^i i^i)$ for $1 \leq i \leq r$.

The algorithm works in an extension ring $\mathsf{R} = \overline{\mathbb{Z}[x]}/(\Gamma)$ of $\mathbb{Z}$, for a monic $\Gamma \in \mathbb{Z}[x]$ of degree $O(\log^2 r)$ with $\log\|\Gamma\| = O(\log^2 r)$. The black-box we use is from Theorem 2.12 and finds the leading minors of $B = PAQ$ for $P \in \mathsf{R}^{r\times m}$ and $Q \in \mathsf{R}^{n\times r}$. The arguments to the black box have height $\|B\| = nm\|A\|\cdot r^{O(1)}$. Thus, we can find all the leading minors of B with $O^\sim(r^{\theta-2}mn\log\|A\| + r^{\theta+1}\log\|A\|)$ bit operations and space for $O^\sim(r^2\log\|A\|)$ bits.

By Theorem 2.8 we require $O((\log(1/\epsilon) + \log\log n + \log\log\|A\|)/\log r)$ evaluations of the black-box plus an additional $O^\sim(r^3\log^2\|A\|)$ bit operations. The total cost is an expected number of $O^\sim((m^\theta n\log\|A\| + r^3\log^2\|A\|)\cdot\log(1/\epsilon))$ bit operations. □

References

A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley (Reading MA), 1974.

E. R. Berlekamp. Factoring polynomials over large finite fields. *Math. Comp.* **24**, pp. 713–735, 1970.

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.

D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comp.* **9**, pp. 251–280, 1990.

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

F. R. Gantmacher. *The Theory of Matrices, Vol. I.* Chelsea Publishing Co. (New York NY), 1990.

M. Giesbrecht. *Nearly Optimal Algorithms for Canonical Matrix Forms.* PhD thesis, University of Toronto, 1993. 196 pp.

G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins University Press (Baltimore, USA), 1983.

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.

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

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, M. S. Krishnamoorthy, and B. D. Saunders. Fast parallel computation of Hermite and Smith forms of polynomial matrices. *SIAM J. Algebraic and Discrete Methods* **8**, pp. 683–690, 1987.

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.

R. Lidl and H. Niederreiter. *Finite Fields*, vol. 20 of *Encyclopedia of Mathematics and its Applications.* Addison-Wesley (Reading MA), 1983.

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

J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Ill. J. Math.* **6**, pp. 64–94, 1962.

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.

Efficient Parallel Solution of Sparse Systems of Linear Diophantine Equations

M. Giesbrecht[†]

Technical Report No. 97/02

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

January 24, 1997

[*]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.

Efficient Parallel Solution of Sparse Systems of Linear Diophantine Equations*

Mark Giesbrecht†

January 24, 1997

Abstract. An efficient new algorithm is presented for solving large sparse systems of linear Diophantine equations which is substantially and provably faster than those previously known in both a sequential and parallel implementation. This is accomplished by reducing the problem of finding an *integer* solution to that of finding a very small number of *rational* solutions of Toeplitz perturbations of the original system. We then employ the Block-Wiedemann algorithm to solve these perturbed systems efficiently in parallel. On an input matrix $A \in \mathbb{Z}^{n\times n}$ of rank r and $w \in \mathbb{Z}^{n\times 1}$, the algorithm finds a $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ with about $O(r(r \log \|A\|_\Delta + \log \|w\|_\Delta)/N)$ matrix-vector products by A modulo single-word primes, on $N \leq r(r \log \|A\|_\Delta + \log \|w\|_\Delta)$ processors. Here $\|A\|_\Delta = \max_{ij} |A_{ij}|$ and $\|w\|_\Delta = \max_i |w_i|$. Additionally, about

$$O\left(r^2 + \frac{rn(r \log \|A\|_\Delta + \log \|w\|_\Delta)}{N} + \frac{n(r \log \|A\|_\Delta + \log \|w\|_\Delta)}{\min(n, N)}\right)$$

bit operations are performed on each processor, ignoring logarithmic factors. With only one processor (i.e., $N = 1$) on a sparse input $A \in \mathbb{Z}^{n\times n}$ with high rank and $O(n^{1+\xi})$ non-zero entries (for some $0 \leq \xi \leq 1$) our new algorithm improves on the cost of the best known sequential algorithm by a factor of almost $n^{1-\xi}$.

1 Introduction

Computing integer solutions to systems of linear Diophantine equations is a classical mathematical problem with many interesting applications in number theory (see Cohen 1993), group theory (see Newman 1972) and combinatorics (see, e.g., Kramer & Mesner 1976). Given an input matrix $A \in \mathbb{Z}^{n\times n}$ and vector $w \in \mathbb{Z}^{n\times 1}$, the problem is to find *integer* vectors $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$. It appears to be considerably harder to compute integer solutions than solutions over $\mathbb{Q}$ or more general fields, the main difficulty being controlling (potentially exponential) intermediate expression swell. Moreover, in practice many of the matrices encountered are sparse (lots of entries are zero) and it is desirable to exploit this in our algorithms (see, e.g., Kramer & Mesner 1976, Hafner & McCurley 1989). For matrices over fields this has been accomplished admirably by the algorithms of Wiedemann (1986),

*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, Manitoba, Canada, R3T 2N2. Email: mwg@cs.umanitoba.ca

Coppersmith (1994) and Kaltofen (1995). The latter two algorithms are also extremely well-suited to a coarse-grained parallel implementation. In this paper we show how to achieve this same success with sparse integer matrices, producing integer solutions of small size while eliminating intermediate expression swell and fill-in. Our algorithm gives a substantial improvement for sparse matrices over the best known algorithms (see below) in both sequential and coarse-grained parallel implementations. The main result we demonstrate is (summarized from Corollary 5.4):

> *Let $A \in \mathbb{Z}^{n\times n}$ with rank r and $w \in \mathbb{Z}^{n\times 1}$, and assume a solution $v \in \mathbb{Z}^{n\times 1}$ to $Av = w$ exists. Let $\varrho = r\log\|A\|_\Delta + \log\|w\|_\Delta$ and suppose we are computing on a network of $N \le r\varrho$ processors.*
>
> - *We can find a $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ with an expected number of $\tilde{O}(r\varrho/N)$ matrix-vector products by A modulo primes with $O(\log n + \log\log(\|A\|_\Delta + \|w\|_\Delta))$ bits.*
> - *The output v satisfies $\log\|v\|_\Delta = \tilde{O}(r\log n + r\log\|A\|_\Delta + \log\|w\|_\Delta)$.*
> - *An additional $\tilde{O}(r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))$ bit operations is executed simultaneously by each processor.*
> - *Each processor requires additional storage for $\tilde{O}(n + n\rho/\min(n,N))$ words (not including possibly shared images of A modulo single-word primes).*

The algorithm is probabilistic of the the Las Vegas type: solutions produced are guaranteed correct and, if a solution exists for a particular input, any invocation of the algorithm on that input produces a solution with probability at least $1/2$. $O(\mathrm{M}(l))$ bit operations are required to multiplying two integers with l bits ($\mathrm{M}(l) = l^2$ with standard arithmetic and $\mathrm{M}(l) = l\log l\log\log l$ using FFT-based methods). For convenience we occasionally use "soft-Oh" notation in our cost analyses: for any $f, g : \mathbb{R}^l \to \mathbb{R}$, $f = \tilde{O}(g)$ if and only if $f = O(g\cdot\log^c g)$ for some constant $c > 0$. This considerably simplifies notation when working with modular algorithms: the primes used typically fit in single (32-bit or 64-bit) machine words and can be operated on at unit cost, but have (unavoidable) logarithmic and doubly logarithmic factors in their length when analysed exactly.

Like the algorithms of Wiedemann (1986) and Coppersmith (1994) which motivated this work, we employ the so-called "black-box" paradigm, in which a matrix is defined by its action on vectors by matrix-vector product. Individual entries of the input matrix are not manipulated directly. Clearly a matrix with lots of zero entries will have a fast black box. As in Giesbrecht (1996) we adapt this technique to integer matrices by working with matrix-vector products modulo word-sized primes. Our goal then is to demonstrate comparable results with Diophantine linear systems as have been obtained for systems over a field.

Early attempts at solving systems of linear Diophantine equations go back at least to Blankinship (1966), Borosh & Fraenkel (1966) and Bradley (1971), while the first polynomial-time solutions appear in Frumkin (1976) and Kannan & Bachem (1979). Since then, there have been many improvements; see, e.g., Chou & Collins (1982), Iliopolous (1989), Havas *et al.* (1993), Havas & Majewski (1994), Storjohann & Labahn (1996) and Storjohann (1996). Most of these methods proceed by computing a triangular (Hermite) or diagonal (Smith) form of A with multiplier matrices, from which the space of solutions to the system is easily determined. Storjohann (1996) presents the best known solution to date:

On input $A \in \mathbb{Z}^{m\times n}$ and $w \in \mathbb{Z}^{m\times 1}$, with $m \le n$, a vector $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ can be found with $O(nm^3 \log^2(\|A\|_\Delta + \|w\|_\Delta) + m^4 \log^3(\|A\|_\Delta + \|w\|_\Delta))$ bit operations using standard integer and matrix arithmetic. The output $v \in \mathbb{Z}^{n\times 1}$ satisfies $\log \|v\|_\Delta = O(m \log(m) \cdot (\log \|A\|_\Delta + \log \|w\|_\Delta))$.

Here $\|A\|_\Delta = \max_{ij} |A_{ij}|$, the Δ-*norm* of A. This is close to the best possible asymptotic cost for dense matrices without resorting to non-standard matrix arithmetic, and is very close to the cost of finding a rational solution to the same system. By comparison, our new algorithm, implemented sequentially ($N = 1$), performs comparably — even marginally better — on *dense* input, and substantially better on *sparse* input:

On input $A \in \mathbb{Z}^{m\times n}$ with $O(nm^\xi)$ non-zero elements (for some $0 \le \xi \le 1$) and $w \in \mathbb{Z}^{m\times 1}$, with $m \le n$, a vector $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ can be found with an expected number of $\tilde{O}(nm^{2+\xi} \log(\|A\|_\Delta + \|w\|_\Delta) + nm^2 \log^2(\|A\|_\Delta + \|w\|_\Delta))$ bit operations using standard integer and matrix arithmetic. The output $v \in \mathbb{Z}^{n\times 1}$ satisfies $\log \|v\|_\Delta = \tilde{O}(m \log n + m \log \|A\|_\Delta + \log \|w\|_\Delta)$.

The basic idea behind our algorithm is to solve the leading $r \times r$ system (where $r = \operatorname{rank} A$) of a small set of Toeplitz perturbations of the original system. Let $U, L \in \mathbb{Z}^{n\times n}$ be random unimodular upper and lower triangular Toeplitz matrices respectively, and consider solving the system $UALv = Uw$. Kaltofen & Saunders (1991) showed that over the rationals the leading $r \times r$ submatrix B_r of UAL is strongly non-singular, and by solving this system we quickly obtain as solution $\hat{v} \in \mathbb{Q}^{n\times 1}$ to $A\hat{v} = w$. In Section 2 we extend Kaltofen & Saunders' result by noting that if $d_1, \ldots, d_r \in \mathbb{Z}$ are the determinantal divisors of A, and p is a "large" prime dividing d_k, then the order of p in the leading $k \times k$ minor of B equals the order of p in d_k with high probability. Moreover, with high probability p does not divide the denominators of any of the coefficients of the obtained solution $\hat{v}$. This is proven by examining the solution manifold of the perturbed system in the p-adic closure $\mathbb{Q}_p$ of $\mathbb{Q}$. By considering a very small number ($\approx \log\log(n + \|A\|_\Delta)$) of perturbed systems we hopefully obtain a series of rational solutions whose denominators are relatively prime, from which we can construct an integer solution vector. We prove that using the above technique we can efficiently find a solution whose coefficients have "smooth" denominators, i.e., only divisible by primes less than $2r(r + 1)$. This method is realized in the algorithm `SmoothSolver` in Section 3.

Unfortunately our analysis fails for small primes dividing d_r (even if the algorithm does not seem to fail often in practice). The problem stems from the failure of the inequality used to bound away from zero the probability of getting a non-zero of a multi-variate polynomial (the so called Zippel-Schwartz Lemma) in this case. To overcome this we considerably extend a technique developed in Giesbrecht (1995) and work in a very small number of algebraic orders of small degree over $\mathbb{Z}$ such that each small prime dividing d_r remains inert in at least one of these orders (the number, degree, and height of these orders is logarithmic in r). While these orders are no longer PID's (and hence much of the mathematical structure characterizing Diophantine solutions no long exists), their localizations at these inert primes *are* PID's and we think of our algorithms as working in these p-adic closures (even when they really just compute in a small number field). We prove that rational solutions obtained by perturbing with Toeplitz matrices over these orders, and solving over their quotient number fields, are free of small primes dividing their denominators with high probability.

The algorithms for generating these orders with specified inert primes, and the theory for working with their localizations is presented in Section 4. Finally, in Section 5 we present an algorithm `RefineToDiophantine` which takes a smooth rational solution and produces a Diophantine solution. The structure of this algorithm is almost identical to that of `SmoothSolver` except for the computation in number fields; the cost is within a poly-logarithmic factor.

Definitions and Notation

We denote by $\mathbb{F}_p$ the finite field with p elements (*not to be confused with the p-adic integers* $\mathbb{Z}_p$, *to be introduced later*).

We define a height function on $\mathbb{Q}$ as follows. For $a, b \in \mathbb{Z}$ with $\gcd(a,b) = 1$, we define the *height* of $a/b \in \mathbb{Q}$ as $\mathcal{H}(a/b) = \max\{|a|, |b|\}$. The Δ-*norm* of a matrix $B \in \mathbb{Q}^{m\times n}$ is defined as $\|B\|_\Delta = \max_{ij} \mathcal{H}(B_{ij})$ and of a polynomial $g = \sum_{0\le i\le m} b_i x^i \in \mathbb{Q}[x]$ as $\|g\|_\Delta = \max_i \mathcal{H}(b_i)$.

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 R, with $B \in \mathsf{R}^{m\times n}$, $\sigma = (b_1, \ldots, b_k) \in \mathcal{C}_k^m$ and $\tau = (c_1, \ldots, c_k) \in \mathcal{C}_k^n$ define the *submatrix* $B\left[\begin{smallmatrix}\sigma\\ \tau\end{smallmatrix}\right]$:

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

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

2 Conditions and perturbations for Diophantine solutions

In this section we present the necessary mathematical underpinnings to our algorithm for solving Diophantine equations. Much of this section is presented abstractly for integral (entire) principal ideal domains. We will typically then apply these theorems to localizations of $\mathbb{Z}$ and more general algebraic orders of number fields.

Smith dominant matrices over principal ideal domains

Let R be an integral principal ideal domain and K its field of fractions. We write $a \sim b$ if there exists a $\mu \in \mathsf{R}^*$ such that $a = \mu b$. Let $B \in \mathsf{R}^{n\times n}$ of rank r with non-zero determinantal divisors $d_1, \ldots, d_r \in \mathsf{R}$. We say that B is *Smith dominant* if $B\binom{1\ldots k}{1\ldots k} \sim d_k$ for $1 \le i \le r$. Note that if R is a field, Smith dominant matrices are exactly those which are strongly non-singular, that is, all leading minors are non-zero.

THEOREM 2.1. *Let $B \in \mathsf{R}^{n\times n}$ be Smith dominant of rank r with non-zero determinantal divisors $d_1, \ldots, d_r$ and $w \in \mathsf{R}^{n\times 1}$. There exists a solution $v \in \mathsf{R}^{n\times 1}$ such that $Bv = w$ if and only if there exist $v_1, \ldots, v_r \in \mathsf{R}$ such that $B(v_1, \ldots, v_r, 0, \ldots, 0)^t = w$.*

REMARK 2.2. *Since $B\binom{1\ldots r}{1\ldots r} \sim d_r \neq 0$, $(v_1, \ldots, v_r)^t$ is the unique solution in $\mathsf{K}^{r\times 1}$ of*

$$B\begin{bmatrix}1\ldots r\\ 1\ldots r\end{bmatrix}\begin{pmatrix} v_1 \\ \vdots \\ v_r \end{pmatrix} = \begin{pmatrix} w_1 \\ \vdots \\ w_r \end{pmatrix}.$$

where $w = (w_1, \ldots, w_n)^t$.

PROOF. Since B is Smith dominant, standard unmodular row and column elimination on B (without pivoting) yields the factorization $B = XSY$, where $X \in \mathsf{R}^{n\times n}$ is lower triangular with ones on the diagonal, Y is upper triangular with ones on the diagonal and $S = \mathrm{diag}(s_1, \ldots, s_r, 0, \ldots, 0) \in \mathsf{R}^{n\times n}$ is the Smith form of B (that is $s_1 \sim d_1$ and $s_i \sim d_i/d_{i-1}$ for $2 \le i \le r$). Then $Bv = w \iff XSYv = w \iff SYv = X^{-1}w \iff S\hat{v} = \hat{w}$, where $\hat{v} = Yv$ and $\hat{w} = X^{-1}w$. Suppose there exists a solution $v \in \mathsf{R}^{n\times 1}$ to $Bv = w$. Then there exists a $\hat{v} \in \mathsf{R}^{n\times 1}$ such that $S\hat{v} = \hat{w}$, and we can choose $\hat{v} = (\hat{v}_1, \ldots, \hat{v}_r, 0, \ldots, 0)^t \in \mathsf{R}^{n\times 1}$ (since columns $r+1 \ldots n$ of S are all zeros). This yields $v = Y^{-1}\hat{v}$ as a solution to $Av = w$ and $v = (y_1, \ldots, y_r, 0, \ldots, 0) \in \mathsf{R}^{n\times 1}$ since Y^{-1} is also upper triangular. The converse is trivial. □

Toeplitz perturbations into Smith dominant form

Let R be an integral principal ideal domain and K its field of quotients. Define

$$\mathfrak{U} = \begin{pmatrix} 1 & x_2 & x_3 & \cdots & x_n \\ & 1 & x_2 & \ddots & \vdots \\ & & 1 & \ddots & x_3 \\ & & & 1 & x_2 \\ & & & & 1 \end{pmatrix}, \quad \mathfrak{L} = \begin{pmatrix} 1 & & & & \\ y_2 & 1 & & & \\ y_3 & y_2 & 1 & & \\ \vdots & \ddots & \ddots & 1 & \\ y_n & \cdots & y_3 & y_2 & 1 \end{pmatrix}$$

where $\Lambda = \{x_2, \ldots, x_{n-1}, y_2, \ldots, y_{n-1}\}$ is a set of algebraically independent indeterminates over K.

THEOREM 2.3. *Let $A \in \mathsf{R}^{n\times n}$ have rank r and $\mathfrak{B} = \mathfrak{U}A\mathfrak{L} \in \mathsf{R}[\Lambda]^{n\times n}$. For $1 \le k \le r$ we have $\mathrm{cont}(\mathfrak{B}\binom{1\ldots k}{1\ldots k}) \sim d_k$, where d_k is the kth determinantal divisor of A.*

PROOF. Using a Binet-Cauchy minor expansion (see Gantmacher 1990, p. 9), we have

$$\mathfrak{B}\binom{1\ldots k}{1\ldots k} = \sum_{\sigma,\tau\in C_k^n} \mathfrak{U}\binom{1\ldots k}{\sigma}\mathfrak{L}\binom{\tau}{1\ldots k} \cdot A\binom{\sigma}{\tau}.$$

Under the variable ordering $x_2 < \cdots < x_n$ and $y_2 < \cdots < y_n$, Kaltofen & Saunders (1991) show that the lexicographically smallest term of $\mathfrak{U}\binom{1\ldots k}{\sigma}$ and $\mathfrak{L}\binom{\tau}{1\ldots k}$ are unique to this choice of σ, τ. Thus the polynomials $f_{\sigma,\tau} = \mathfrak{U}\binom{1\ldots k}{\sigma}\mathfrak{L}\binom{\tau}{1\ldots k} \in \mathsf{R}[\Lambda]$ are linearly independent over K, and in fact over any quotient field $\mathsf{R}/p\mathsf{R}$ for any prime $p \in \mathsf{R}$. Let p be a prime in R and $l = \mathrm{ord}_p(d_k)$. Clearly, $p^l \mid \mathrm{cont}(\mathfrak{B}\binom{1\ldots k}{1\ldots k})$. Suppose $p^{l+1} \mid \mathrm{cont}(\mathfrak{B}\binom{1\ldots k}{1\ldots k})$. Then

$$\sum_{\sigma,\tau\in C_k^n} \mathfrak{U}\binom{1\ldots k}{\sigma}\mathfrak{L}\binom{\tau}{1\ldots k} \cdot A\binom{\sigma}{\tau}/p^l = \sum_{\sigma,\tau\in C_k^n} f_{\sigma,\tau} \cdot A\binom{\sigma}{\tau}/p^l \equiv 0 \bmod p.$$

This implies the $f_{\sigma,\tau}$'s are linearly dependent modulo p or that $A\binom{\sigma}{\tau} \equiv 0 \bmod p^{l+1}$ for all $\sigma, \tau \in C_k^n$. The latter statement is false by our definition of l, and the former leads to a contradiction. Thus $\mathrm{ord}_p d_k = \mathrm{ord}_p \mathrm{cont}(\mathfrak{B}\binom{1\ldots k}{1\ldots k})$ for all $p \in \mathsf{R}$, whence $d_k \sim \mathrm{cont}(\mathfrak{B}\binom{1\ldots k}{1\ldots k})$. □

We can use the above theorem to perturb a matrix into Smith dominant form with high probability. We will employ the "Zippel-Schwartz" lemma to bound the probability of obtaining a zero of a multi-variate polynomial:

FACT 2.4 *(Zippel 1979, Schwartz 1980). Assume* $f \in \mathsf{D}[x_1, \ldots, x_k]$ *is non-zero,* D *an integral domain, and* $\mathcal{V}$ *a finite subset of* D. *Suppose elements* $a_1, \ldots, a_k$ *are randomly and uniformly chosen from* $\mathcal{V}$. *Then* $\text{Prob}\{f(a_1, \ldots, a_k) = 0 : a_1, \ldots, a_k \in \mathcal{V}\} \leq \deg(f)/\#\mathcal{V}$.

THEOREM 2.5. *Let* $A \in \mathsf{R}^{n\times n}$ *with rank* r *and determinantal divisors* $d_1, \ldots d_r \in \mathsf{R}$. *Let* $p \in \mathsf{R}$ *a prime in* R *and* $\mathcal{V}$ *a finite subset of* R *whose elements are in distinct cosets modulo* p. *Suppose* $u_2, \ldots, u_n, l_2, \ldots, l_n$ *are chosen randomly and uniformly from* $\mathcal{V}$ *and we construct*

$$B = UAL, \text{ where } U = \begin{pmatrix} 1 & u_2 & u_3 & \cdots & u_n \\ & 1 & u_2 & \ddots & \vdots \\ & & 1 & \ddots & u_3 \\ & & & 1 & u_2 \\ & & & & 1 \end{pmatrix} \quad L = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ l_3 & l_2 & 1 & & \\ \vdots & \ddots & \ddots & 1 & \\ l_n & \cdots & l_3 & l_2 & 1 \end{pmatrix} \tag{2.1}$$

Then

$$\text{Prob}\left\{ \text{ord}_p B\binom{1\ldots k}{1\ldots k} = \text{ord}_p d_k \quad \forall k : 1 \leq k \leq r \right\} \geq 1 - \frac{r(r+1)}{\#\mathcal{V}}.$$

PROOF. For any k,

$$B\binom{1\ldots k}{1\ldots k} = d_k \cdot f_k(u_2, \ldots, u_n, l_2, \ldots, l_n)$$

for some $f_k \in \mathsf{R}[x_2, \ldots, x_n, y_2, \ldots, y_n]$ with content 1 and degree $2k$ by Theorem 2.3. Thus p has the same order in $B\binom{1\ldots k}{1\ldots k}$ as in d_k if and only if $f_k(u_2, \ldots, u_n, l_2, \ldots, l_n) \not\equiv 0 \bmod p$. Since all elements of $\mathcal{V}$ are in distinct cosets modulo p, by Fact 2.4, $f_k(u_2, \ldots, u_n, l_2, \ldots, l_n) \equiv 0 \bmod p$ with probability at most $2k/\#\mathcal{V}$. Thus the probability of $f_k(u_2, \ldots, u_n, l_2, \ldots, l_n) \equiv 0 \bmod p$ for any $1 \leq k \leq r$ is at most $\sum_{1\leq k\leq r}(2k)/\#\mathcal{V} = r(r+1)/\#\mathcal{V}$. □

The following simple lemma allows us to solve a perturbed system to obtain a solution to the original system.

LEMMA 2.6. *Let* $A \in \mathsf{R}^{n\times n}$ *and* $w \in \mathsf{R}^{n\times 1}$. *Let* $U, L \in \mathsf{R}^{n\times n}$ *with* $\det U, \det L \in \mathsf{R}^*$ *and* $B = UAL$. *Then* $\bar{v} \in \mathsf{R}^{n\times 1}$ *is a solution to* $B\bar{v} = Uw$ *if an only if* $v = L\bar{v}$ *is a solution to* $Av = w$.

PROOF. For the forward direction, assume $\bar{v}$ is a solution to $B\bar{v} = Uw$. Then

$$B\bar{v} = Uw \Longrightarrow UAL\bar{v} = Uw \Longrightarrow AL\bar{v} = w \Longrightarrow Av = w,$$

since U is invertible in $\mathsf{R}^{n\times n}$. Conversely, if $AL\bar{v} = w$ then $UAL\bar{v} = UAL\bar{v} = B\bar{v} = Uw$. □

Localizations of $\mathbb{Z}$ and $\mathbb{Q}$

It will be convenient to consider the localizations of $\mathbb{Q}$ and algebraic number fields at a prime p. We identify the p-adic integers $\mathbb{Z}_p$ and p-adic rationals $\mathbb{Q}_p$ with the (infinite) Laurent series

$$\mathbb{Z}_p = \left\{ \sum_{0 \le i < \infty} a_i p^i : a_i \in \{0, \ldots, p-1\} \right\}, \quad \mathbb{Q}_p = \left\{ \sum_{m \le i < \infty} a_i p^i : a_i \in \{0, \ldots, p-1\},\ m \in \mathbb{Z} \right\}$$

under the usual arithmetic. A useful references for general localizations is Lang (1986), and for p-adic numbers and analysis is Cassels (1986). Clearly $\mathbb{Z}_p \subseteq \mathbb{Q}_p$ and $\mathbb{Z} \subseteq \mathbb{Z}_p$. Also, if $v \in \mathbb{Z}$ is relatively prime with p then $1/v \in \mathbb{Z}_p$ by Hensel's Lemma (essentially p-adic Newton iteration - see Cassels (1986), Lemma 3.1). Since any element in $\mathbb{Q}$ can be written as $p^e u/v$, where $e \in \mathbb{Z}$, $u, v \in \mathbb{Z}$ and $\gcd(v, p) = 1$, we see that $\mathbb{Q} \subseteq \mathbb{Q}_p$. If $a = \sum_{m \le i < \infty} a_i p^i \in \mathbb{Q}_p$ for $a_i \in \{0, \ldots, p-1\}$ and $a_m \neq 0$, we define the *p-adic order* of a as $\mathrm{ord}_p(a) = m$ and the *p-adic norm* of a as $|a|_p = p^{-m}$. with $|0|_p = 0$. Thus $\mathbb{Z}_p = \{a \in \mathbb{Q}_p : |a|_p \le 1\}$.

Parallel modular computation over $\mathbb{Q}$

We next summarize for convenience a standard homomorphic scheme for parallel computing over $\mathbb{Q}$ (see Wang *et al.* 1982, Collins & Encarnación 1995). Let $\Psi : \mathbb{Q}^s \to \mathbb{Q}^t$ be a function we wish to compute and suppose that we know a quickly computable ("upper bound") function $\tau : \mathbb{Q}^s \to \mathbb{R}$ such that $\tau(\bar{x}) \ge \max\{\|\bar{x}\|_\Delta, \|\Psi(\bar{x})\|_\Delta\}$; the cost of computing τ will assumed to be dominated by that of other computations. Suppose also that for all primes $p \in \mathbb{Z}$, except for those in a finite set $\mathcal{B} \subset \mathbb{Z}$, we can compute $\Psi(\bar{x}) \bmod p$ from input $(\bar{x} \bmod p)$ with $O(\psi(s))$ operations in $\mathbb{F}_p$; when $p \in \mathcal{B}$ we can report this fact in the same amount of time. For convenience we will assume that $\#\mathcal{B} = (\log(\tau(\bar{x}))^{O(1)}$.

Following standard practice, we first construct a set $\mathcal{P} \subseteq \mathbb{Z}$ of sufficiently many small primes. We then compute $\Psi(\bar{x}) \bmod p$ for randomly chosen $p \in \mathcal{P}$, rejecting bad primes as we encounter them. Finally, when the product of the good primes chosen is at least $2\tau(\bar{x})^2$, we recover the solution by the Chinese remainder theorem and integer Padé approximation (this is sufficiently many to recover numerator, denominator and sign). See Wang *et al.* (1982). We crudely estimate that at least $\varrho \le \log_2(2\tau(\bar{x})^2)$ good primes are required, though much better estimates are easily computed at run-time.

It is also convenient to allow for an n-point FFT to be performed efficiently (so we may practically multiply polynomials of degree up to n with $O(n \log n)$ operations). To facilitate this, we choose primes p such that $2^l \mid (p-1)$, where $l \ge \lceil \log_2 n \rceil$. By Dirichlet's density theorem on primes in an arithmetic progression, it is easily derived that we can efficiently construct a set $\mathcal{P}$ with $\#\mathcal{P} \ge 2(\#\mathcal{B}) + \varrho$ such that $\log p = O(\log n + \log(\#\mathcal{B}) + \log\log\tau(\bar{x}))$ for all $p \in \mathcal{P}$ (see, e.g., Giesbrecht 1996, Section 3.2). For notational convenience we assume that $n = s^{O(1)}$. From a practical point of view, primes of this size should fit into a single (32-bit or 64-bit) machine word, and operations modulo a prime will have constant cost.

A randomly chosen prime (without replacement) will be bad with probability at most $1/2$. Thus we expect to compute $\Psi(\bar{x}) \bmod p$ for $2\varrho = O(\log(\tau(\bar{x})))$ primes p, and the computation in $\mathbb{F}_p$ requires $O(\psi(s) \cdot (\log n + \log(\#\mathcal{B}) + \log\log\tau(\bar{x}))^2)$ bit operations. Reduction of $\bar{x} \bmod p$ for the used primes $p \in \mathcal{P}$ requires $O^\sim(s \log \|\bar{x}\|_\Delta \cdot \log(\tau(\bar{x})))$ bit operations and recovery of the final integer solution require $O^\sim(t \cdot \mathrm{M}(\log\tau(\bar{x})))$ bit operations; see Wang *et al.* (1982).

We summarize the sequential cost in the following theorem.

THEOREM 2.7. *We can construct a Las Vegas algorithm which on any input $\bar{x} \in \mathbb{Q}^s$ computes $\Psi(\bar{x}) \in \mathbb{Q}^t$. The algorithm requires an expected number of $O^{\sim}(\psi(s) \cdot \log(\tau(\bar{x})) + s\log(\|\bar{x}\|_\Delta)\log(\tau(\bar{x})) + t \cdot \mathrm{M}(\log(\tau(\bar{x}))))$ bit operations. We may assume in our cost function ψ the availability of a practical n-point FFT at cost $O(n\log n)$, where $n = s^{O(1)}$.*

The computation modulo individual primes is independent and hence can be parallelized in a straightforward manner. The three stages of the algorithm, (i) reduction modulo the prime base, (ii) local computation, and (iii) recovery of global solutions, are analysed separately.

THEOREM 2.8. *We can construct a Las Vegas algorithm which on any input $\bar{x} \in \mathbb{Q}^s$ computes $\Psi(\bar{x}) \in \mathbb{Q}^t$ which runs in parallel on N processors:*

(i) for $N \leq s\varrho$, we can reduce $\bar{x}$ mod p for the expected number of ϱ primes used from $\mathcal{P}$ with $O^{\sim}(s\varrho\log\|\bar{x}\|_\Delta/N)$ bit operations carried out simultaneously by each processor;

(ii) for $N \leq \varrho$, we can compute $\Psi(\bar{x} \bmod p)$ for the expected number of ϱ primes p from $\mathcal{P}$ in an expected number of $O^{\sim}(\phi(s) \cdot \varrho/N)$ bit operations carried out simultaneously by each processor;

(iii) for $N \leq t$ we can recover the solutions in $\mathbb{Q}^t$ from images modulo ϱ good primes in an expected number of $O^{\sim}(t \cdot \mathrm{M}(\log(\tau(\bar{x})))/N)$ bit operations carried out simultaneously by each processor;

where $\varrho = \log(\tau(\bar{x}))$. We may assume in our cost function ψ the availability of a practical n-point FFT at cost $O(n\log n)$, where $n = s^{O(1)}$.

3 Finding rational solutions with smooth denominators

We present our algorithm for finding integers solutions to systems of integer equations in two parts. The first part is the basic algorithm and finds a rational solution whose denominators are λ-smooth, that is, only primes less than or equal to λ divide the denominators of the coefficients. This algorithm appears to work well even with $\lambda = 1$ (and hence obtains integer solutions), but unfortunately we can only prove it for $\lambda \geq 2r(r+1)$, where r is the rank of the input matrix. A modification is then presented to deal with the remaining case in a theoretically sound way at an additional logarithmic factor in the cost.

For a $v \in \mathbb{Q}^{n\times 1}$ we define the *denominator* of v to be $\mathrm{denom}(v) = \min\{d \in \mathbb{Z}_{>0} : dv \in \mathbb{Z}^{n\times 1}\}$, the least common multiple of all the denominators of the coefficients (in lowest terms) of v. For any $\lambda > 0$, we say that an integer b is *λ-smooth* if all prime factors of b are less than or equal to λ (or $b = \pm 1$ if $\lambda = 1$).

Our algorithm also has two additional parameters aside from A and w:

- $\lambda > 0$: the returned solution should have a denominator which is λ-smooth. By setting $\lambda = 1$ we achieve integer solutions.
- $\epsilon > 0$: an error tolerance. If it is reported that "No Integer Solution Exists" then this is correct with probability at least $1 - \epsilon$. If a solution is returned, it is always correct. The need for such an error tolerance parameter ϵ is also present in the underlying Wiedemann and Block-Wiedemann algorithms for solving sparse singular systems over a field.

Algorithm: `SmoothSolver`

Input: - $A \in \mathbb{Z}^{n\times n}$ and $w \in \mathbb{Z}^{n\times 1}$;
- a smoothness bound $\lambda > 0$;
- an error tolerance $\epsilon > 0$;

Output: - $v \in \mathbb{Q}^{n\times 1}$ where denom(v) is λ-smooth, or a report "No Integer Solution Exists";

(1) Compute $r := \text{rank}(A)$, correct with probability at least $1 - \epsilon/2$;
(2) $\beta := 2r(r+1)$; $\mathcal{V} := \{-\beta/2, \ldots, \beta/2\} \subseteq \mathbb{Z}$;
$g := 0$;
(3) For $b := 1$ to $\lceil 1 + \log_2(1/\epsilon)\rceil$ Do
(4) For $i := 0$ to $s := \lceil 1 + \log_2(\log_\lambda(n^2\beta^2\|A\|_\Delta))\rceil$ Do
(5) Choose random $u_2, \ldots, u_n, l_2, \ldots, l_n \in \mathcal{V}$;
"Build" a black box for $B = UAL$ where U, L are as in (2.1); Let $B_r = B\left[\begin{smallmatrix}1\ldots r\\ 1\ldots r\end{smallmatrix}\right]$;
$\bar{w} := Uw = (\bar{w}_1, \ldots, \bar{w}_n)^t \in \mathbb{Z}^{n\times 1}$;
(6) Solve $B_r\bar{v} = (\bar{w}_1, \ldots, \bar{w}_r)^t$ for $\bar{v} = (\bar{v}_1, \ldots, \bar{v}_r) \in \mathbb{Q}^{r\times 1}$ with black box for B
If B_r is singular, goto (5);
(7) $v^{(i)} := L(\bar{v}_1, \ldots, \bar{v}_r, 0, \ldots, 0)^t \in \mathbb{Q}^{n\times 1}$; $\delta_i := \text{denom}(v^{(i)})$;
If $Av^{(i)} \neq w$ then report "No solution to Diophantine system exists";
(8) $g := \gcd(g, \delta_i)$;
End For;
End For;
If g is λ-smooth Then
(9) Find $\gamma_0, \ldots, \gamma_s \in \mathbb{Z}$ such that $\sum_{0\le i\le s} \gamma_i\delta_i = g$;
(10) Return $v := (1/g) \cdot \sum_{0\le i\le s} \gamma_i\delta_i \cdot v^{(i)}$;
Else Report "No solution to Diophantine system exists".
End If;

THEOREM 3.1. *The algorithm* `SmoothSolver` *works as specified. Suppose the input matrix* $A \in \mathbb{Z}^{n\times 1}$ *has (unknown) rank* r.

(i) If a solution $v \in \mathbb{Q}^{n\times 1}$ *is returned, it is always correct;*

(ii) $\log\|v\|_\Delta = \tilde{O}(r\log n + r\log\|A\|_\Delta + \log\|w\|_\Delta)$;

(iii) if $\lambda \ge 2r(r+1)$ *and a* λ*-smooth solution exists to the system, a* λ*-smooth solution is found with probability at least* $1 - \epsilon$.

PROOF. The rank of A is obtained with probability at least $1 - \epsilon/2$ via the algorithm of Kaltofen & Saunders (1991) as generalized to integer matrices in Giesbrecht (1996).

For part (i), we note that $A(\delta_i v^{(i)}) = \delta_i w$ for $0 \le i \le s$. Thus

$$Av = A\left((1/g)\cdot\sum_{1\le i\le s}\gamma_i\delta_i\cdot v^{(i)}\right) = \left((1/g)\cdot\sum_{1\le i\le s}\gamma_i\delta_i\right)\cdot w = w$$

and denom$(v) = g$, which is λ-smooth by construction.

For parts (ii) and (iii), first consider an iteration of the inner loop (4)-(8). We have

$$\|B\|_\Delta \le n^2\|U\|_\Delta\cdot\|A\|_\Delta\cdot\|L\|_\Delta \le n^2\beta^2\|A\|_\Delta = O(n^2r^4\|A\|_\Delta),$$
$$\|Uw\|_\Delta \le n\|U\|_\Delta\cdot\|w\|_\Delta \le n\beta\|w\|_\Delta = O(nr^2\|w\|_\Delta).$$

Applying Hadamard's bound and Cramer's rule we find

$$\begin{aligned}
\log_2 |\delta_i| &= O(r \log n + r \log \|A\|_\Delta), \\
\log \|\bar{v}\|_\Delta &= O(r \log r + r \log \|B_r\|_\Delta + \log \|\bar{w}\|_\Delta) = O(r \log n + r \log \|A\|_\Delta + \log \|w\|_\Delta), \\
\log \|v^{(i)}\|_\Delta &= O(r \log n + r \log \|A\|_\Delta + \log \|w\|_\Delta),
\end{aligned}$$

for $0 \le i \le s$. Also, $\log_\lambda \delta_i \le \log_\lambda(n^2\beta^2\|A\|_\Delta)$ is a (crude) upper bound on the number of primes greater than λ which can divide δ_i.

Assume that the rank r is calculated correctly in step (1). Since $B\binom{1\ldots r}{1\ldots r}$ is a non-zero polynomial in $u_2, \ldots, u_n, l_2, \ldots, l_n$ of degree $2r$, B_r is non-singular with probability at least $1 - 2r/(2r(r+1)) = r/(r+1)$ by Fact 2.4. Thus we expect to execute steps (5) and (6) a constant number of times for each iteration of the inner For loop. If a solution to $Av = w$ exists over $\mathbb{Z}$ it certainly exists over $\mathbb{Q}$, and by Lemma 2.6 $v^{(i)}$ will be such a solution (since $\mathbb{Q}$ is a PID). Once the GCD of the denominators is λ-smooth, we execute steps (9) and (10). Step (9) is probably best done in practice by the algorithm of Majewski & Havas (1995), but for a simpler analysis here we employ the algorithm Iliopolous (1989) which finds $\gamma_0, \ldots, \gamma_s \in \mathbb{Z}$ such that $\log |\gamma_i| = O(\log \max_{0 \le i \le s} |\delta_i| \cdot \log s)$. The constructed v thus satisfies

$$\begin{aligned}
\log \|v\|_\Delta &= \log \left(\max \left\{ |g|, \sum_{0 \le i \le s} \gamma_i \|\delta_i v^{(i)}\|_\Delta \right\} \right) \\
&= O((r \log n + r \log \|A\|_\Delta)(\log\log\log n + \log\log\log \|A\|_\Delta) + \log \|w\|_\Delta)
\end{aligned}$$

or $O(r \log n + r \log \|A\|_\Delta + \log \|w\|_\Delta)$, which proves (ii).

To prove (iii) assume that an λ-smooth solution does indeed exist. We show that with each iteration of the outer For loop, the algorithm finds such a solution with probability at least $1/2$. Let $p > \lambda$ be a prime dividing δ_0. Since $\#(\mathcal{V} \bmod p) \ge 2r(r+1)$, by Theorem 2.5,

$$\text{Prob}\left\{ \text{ord}_p B\binom{1\ldots k}{1\ldots k} = \text{ord}_p d_k \quad \forall k : 1 \le k \le r \right\} \ge 1/2.$$

If indeed $\text{ord}_p B\binom{1\ldots k}{1\ldots k} = \text{ord}_p d_k$ for all k $(1 \le k \le r)$, the image of B in $\mathbb{Z}_p^{n \times n}$ is Smith dominant. Thus by Theorem 2.1, the image of $\bar{v}$ in $\mathbb{Q}_p^{r \times 1}$ lies in $\mathbb{Z}_p^{r \times 1}$ and the image of $v^{(i)}$ in $\mathbb{Q}_p^{n \times 1}$ lies in $\mathbb{Z}_p^{n \times 1}$, whence $p \nmid \text{denom}(v^{(i)})$. Thus, the probability that $p \mid \text{denom}(v^{(i)})$ for all $1 \le i \le s = 1 + \log_2(\log_\lambda(n^2\beta^2\|A\|_\Delta))$ is at most $(1/2) \cdot 1/\log_\lambda(n^2\beta^2\|A\|_\Delta)$. The probability this is true for *any* prime $p \ge \lambda$ dividing δ_0 is thus at most $1/2$, since there are at most $\log_\lambda(n^2\beta^2\|A\|_\Delta)$ such primes. By executing the outer For loop $1 + \log_2(1/\epsilon)$ times we ensure that if a solution exists (and we obtained the rank correctly), we will find a solution with probability at least $1 - \epsilon/2$. Since the rank is correct with probability $1 - \epsilon/2$, the theorem follows. □

We will employ the Wiedemann and Block-Wiedemann linear equation solvers over a finite field, as developed in Wiedemann (1986), Kaltofen & Saunders (1991) and Coppersmith (1994), and analysed in Kaltofen (1995).

FACT 3.2. *Suppose we are given a black box for a non-singular matrix $B \in \mathsf{K}^{r\times r}$ and vector $\bar{w} \in \mathsf{K}^{r\times 1}$ over a field K with at least $16r^2$ elements. On a network of $N \le r$ processors we can solve $B\bar{v} = \bar{w}$ for $\bar{v} \in \mathsf{K}^{r\times 1}$ with an expected $O(r/N)$ matrix-vector products by B and $O(r^2 \log r)$ operations in K, executed simultaneously on each processor (assuming an r-point FFT is available in K). Each processor requires additional storage for $O(r)$ elements of K (not including a possibly shared image of B).*

This algorithm can be applied to non-singular rational matrices as a direct application of the techniques of Theorem 2.8. See Kaltofen & Saunders (1991) for a different approach.

THEOREM 3.3. *Suppose we are given a black box for a non-singular matrix $B \in \mathbb{Z}^{r\times r}$ and vector $\bar{w} \in \mathbb{Z}^{r\times 1}$ and wish to solve $B\bar{v} = \bar{w}$ for $\bar{v} \in \mathbb{Q}^{r\times 1}$. Let $\varrho = r \log \|B\|_\Delta + \log \|\bar{w}\|_\Delta$. On a network of $N \le r\varrho$ processors we can solve for $\bar{v}$ with an expected $O\tilde{}(r\varrho/N)$ matrix-vector products by B modulo (single-word) primes with $O(\log r + \log\log(\|B\|_\Delta + \|\bar{w}\|_\Delta))$ bits. An additional $O\tilde{}(r^2 + r\,\mathrm{M}(\varrho)/\min(r,N))$ bit operations is executed simultaneously by each processor. Each processor requires additional storage for $O(r\rho/\min(r,N))$ words (not including possibly shared images of B modulo single-word primes).*

PROOF. To apply Theorem 2.8, we need only note that the only bad primes are those which divide the determinant of B, and there are at most $O(r(\log r + \log \|B\|_\Delta))$. It is also generally convenient to eliminate small primes (say those less than $16r^2$) to allow the Wiedemann and Block-Wiedemann algorithms to (provably) work without the use of field extensions. □

We are parallelizing the linear solver in two different ways. First, we break the problem into an expected ϱ independent problems modulo ϱ distinct primes. Second, for each prime we use up to r processors to solve a non-singular system over a finite field via the Block-Wiedemann algorithm. A potential bottleneck is the recovery of rational solutions: each of the r entries in the solution vectors is recovered independently from its modular images on up to r processors. If $\mathrm{M}(\varrho) = \varrho^2$ then the recovery phase potentially dominates the overall cost, at least in theory.

THEOREM 3.4. *Let $A \in \mathbb{Z}^{n\times n}$ of (unknown) rank $r \le m$, $w \in \mathbb{Z}^{n\times 1}$, $\lambda > 0$ and $\epsilon > 0$ be as in the input to* `SmoothSolver`. *Let $\varrho = r \log \|A\|_\Delta + \log \|w\|_\Delta$ and suppose we are computing on a network of $N \le r\varrho$ processors.*

(i) If a λ-smooth solution $v \in \mathbb{Q}^{n\times 1}$ to $Av = w$ exists, `SmoothSolver` *finds one with an expected number of $O\tilde{}(r\varrho/N)$ matrix-vector products by A modulo primes with $O(\log n + \log\log(\|A\|_\Delta + \|w\|_\Delta))$ bits. An additional $O\tilde{}(r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))$ bit operations is executed simultaneously by each processor.*

(ii) If no λ-smooth solution $v \in \mathbb{Q}^{n\times 1}$ to $Av = w$ exists, `SmoothSolver` *requires an expected number of $O\tilde{}((r\varrho/N)\cdot\log(1/\epsilon))$ matrix-vector products by A modulo primes with $O(\log n + \log\log(\|A\|_\Delta + \|w\|_\Delta))$ bits. An additional $O\tilde{}((r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))\cdot \log(1/\epsilon))$ bit operations is executed simultaneously by each processor.*

Each processor requires storage for an additional $O(n + n\rho/\min(n,N))$ words (not including possibly shared images of A modulo single-word primes).

PROOF. The inner For loop iterates $O(\log\log n + \log\log \|A\|_\Delta)$ times. If a solution exists, we expect the outer loop to iterate twice. If no solution exists, the outer For loop iterates $O(\log(1/\epsilon))$ times.

Each evaluation of the black box for $y \to B_r y$ where $y = (y_1, \ldots, y_r)^t \in \mathbb{Z}^{r\times 1}$ is performed by evaluating $UAL(y_1, \ldots, y_r, 0, \ldots, 0)^t = (z_1, \ldots, z_n)^t$, and returning $(z_1, \ldots, z_r)^t = B_r y$. Pre-multiplication by a unit triangular Toeplitz matrix takes $O(n \log n)$ operations in the ground field assuming an n-point FFT (see Kailath (1980)). Thus each matrix-vector product by B_r requires one black box evaluation of A modulo primes with $O(\log r + \log\log(\|B\|_\Delta + \|\bar{w}\|_\Delta))$ or $O(\log n + \log\log(\|A\|_\Delta + \|w\|_\Delta))$ bits, plus $O(n \log n)$ additional operations modulo primes of this same size. The linear system $B_r \bar{v} = \bar{w}$ in step (6) is then solved using the Block-Wiedemann method described in Theorem 3.3. The algorithm Iliopolous (1989), which finds $\gamma_0, \ldots, \gamma_c \in \mathbb{Z}$, requires $O((\log\log n + \log\log \|A\|_\Delta) \cdot (\log r + \log\log n + \log\log \|A\|_\Delta) \cdot \mathrm{M}(r \log n + r \log \|A\|_\Delta))$ or $O\tilde{}(\mathrm{M}(r \log \|A\|_\Delta))$ bit operations, which we will execute on a single processor. Finally, to recover the solutions in $\mathbb{Z}^{n\times 1}$ requires $O\tilde{}(n\,\mathrm{M}(\varrho))$ to the Chinese remainder algorithm and integer Padé approximation on each coefficient (see Bach & Shallit 1996). □

4 Constructing algebraic orders with selected inert primes

The main theoretical hurdle to be overcome in finding Diophantine solutions (instead of just solutions with smooth denominators) is that the Zippel-Schwartz lemma fails us for small primes dividing the determinantal divisors. Our solution is to work in a collection of small extension rings over $\mathbb{Z}$. Recall that an *algebraic order*, or simply an *order*, is a submodule of the ring of integers of a number field (see, e.g., Cassels (1986), Chapter 10) and contains $\mathbb{Z}$ as a subring. In this section we describe how to construct orders of number fields such that certain primes remain inert (i.e., the ideals they generate remain prime) of some prescribed degree. We also discuss some useful properties of the p-adic integral closures of these orders which will be important in the next section.

For any $\eta \in \mathbb{N}$, and $s \in \mathbb{R}_{>0}$ define $\mathcal{M}(\eta; s) = \{g \in \mathbb{Z}[z] : g \text{ monic}, \deg g = \eta, \|g\|_\Delta \le s\}$.

Algorithm: `BuildOrders`
Input: – $\eta \in \mathbb{Z}$ and primes $p_1, \ldots, p_\kappa \in \{2, \ldots, \tau\}$;
Output: – a set $G \subseteq \mathcal{M}(\eta; \eta\tau)$ such that for each $i \in \{1, \ldots, \kappa\}$, there exists a $\Gamma_i \in G$ with $\Gamma_i \bmod p_i$ irreducible in $\mathbb{F}_{p_i}[z]$.

(1) Repeat
(2) Let $\mathcal{P} := \{1, \ldots, \kappa\}$; $G := \{\}$;
(3) Let $l := 8\eta \log(2\kappa)$;
(4) For $j := 1$ to l while $\mathcal{P} \neq \{\}$ do
(5) Choose a random $h_j \in \mathcal{M}(\eta; \eta\tau)$;
(6) For any $i \in \mathcal{P}$ do
(7) If $h_j \bmod p_i \in \mathbb{F}_{p_i}[z]$ is irreducible in $\mathbb{F}_{p_i}[z]$
Then $\mathcal{P} := \mathcal{P} \setminus \{i\}$; $G := G \cup \{h_j\}$;
End For;
End For;
Until $\mathcal{P} = \{\}$;
(8) Return G.

THEOREM 4.1. *The algorithm* `BuildOrders` *always produces the correct results as described and requires an expected number of* $O((\eta^3 + \eta^2 \log \tau) \cdot \kappa\eta \log \kappa \cdot \log^2 \tau)$ *bit operations. The number of polynomials in* G *is* $O(\eta \log \kappa)$ *and each has* Δ*-norm at most* $\eta\tau$.

PROOF. First, for any prime p and $\eta \in \mathbb{N}$, define

$$\mathcal{M}_p(\eta) = \{g \in \mathbb{F}_p[z] : g \text{ monic}, \deg g = \eta\} = \mathcal{M}(\eta; \eta\tau) \bmod p.$$

For a randomly chosen $h \in \mathcal{M}_p(\eta)$ and $\eta \geq 3$, the probability that h is irreducible in $\mathbb{F}_p[z]$ is at least

$$\frac{1}{\eta} \sum_{d \mid \eta} \mu(d) q^{\eta/d} \geq \frac{p^\eta}{\eta} - \frac{p(p^{\eta/2} - 1)}{\eta(p-1)} \geq \frac{3p^\eta}{4\eta}$$

by Lidl & Niederreiter (1983), Exercise 3.27. If we choose h randomly and uniformly from $\mathcal{M}(\eta; \eta\tau)$, $h \bmod p$ falls into any particular residue class in $\mathcal{M}_p(\eta)$ with probability at least $(\lfloor (2\eta\tau + 1)/p \rfloor / (2\eta\tau+1))^\eta \geq (1/p - 1/(2\eta\tau+1))^\eta$. The probability that $h \bmod p$ is irreducible in $\mathbb{F}_p[z]$ is at least

$$\left(\frac{1}{p} - \frac{1}{2\eta\tau + 1}\right)^\eta \cdot \frac{3p^\eta}{4\eta} = \frac{3}{4\eta} \cdot \left(1 - \frac{p}{2\eta\tau + 1}\right)^\eta > \frac{3}{4\eta} \cdot \left(1 - \frac{1}{2\eta}\right)^\eta \geq \frac{3}{8\eta}.$$

For any fixed prime $p_i \in \{p_1, \ldots, p_\kappa\}$, the probability that in a single iteration of the outer loop steps (2)-(7), for all random choices in step (5), we do not choose an $h_j \in \mathcal{M}(\eta; \eta\tau)$ with $h_j \bmod p_i$ irreducible in $\mathbb{F}_{p_i}[z]$ is at most $(1 - 3/(8\eta))^l$. The probability that there exists *any* prime $p_i \in \{p_1, \ldots, p_\kappa\}$ for which we do not choose such an h_j is thus at most $\kappa \cdot (1 - 3/(8\eta))^l < 1/2$ by our choice of $l = 8\eta \log(2\kappa)$.

For each random choice of $h_i \in \mathcal{M}(\eta; \eta\tau)$ the inner loop of steps (6)–(7) can be accomplished with an expected number of $O((\eta^3 + \eta^2 \log \tau) \cdot \kappa \cdot \log^2 \tau)$ bit operations using Berlekamp's (1970) factoring algorithm, and this loop is executed $l = 8\eta \log(2\kappa)$ times per iteration of the outer loop. □

Heights and localizations of algebraic orders

Let $\Gamma = \sum_{0 \leq i \leq \eta} \gamma_i z^i \in \mathbb{Z}[z]$ be monic and irreducible of degree η and $\theta = z \bmod \Gamma$, so $\mathbb{Z}[\theta] = \mathbb{Z}[z]/(\Gamma)$ is an algebraic order and a sub-order of the maximal order $\mathcal{O}$ (the ring of algebraic integers) in $\mathbb{Q}(\theta)$. Computationally we represent $\mathbb{Z}[\theta]$ with respect to the power basis $\{1, z, z^2, \ldots, z^{\eta-1}\}$, where elements are uniquely represented by an integer polynomial of degree less than η (under standard addition and multiplication of polynomials, reduced modulo Γ).

We define a *Height* function $\mathcal{H} : \mathbb{Q}(\theta) \to \mathbb{N}$ as follows. Let $\Theta \in \mathbb{Z}^{\eta \times \eta}$ be the companion matrix of Γ. For $a = \sum_{0 \leq i < \eta} a_i \theta^i \in \mathbb{Z}[\theta]$, define $\mathcal{H}(a) = \| \sum_{0 \leq i < \eta} a_i \Theta^i \|_\infty$. It is easily verified that

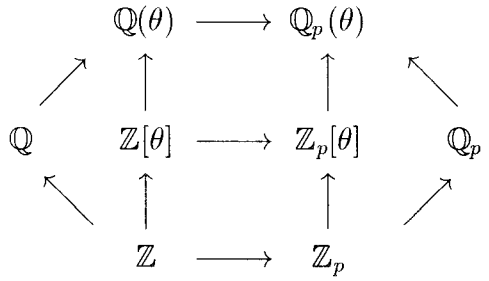
$$\mathcal{H}(a) \leq \begin{cases} |a| & \text{if } a \in \mathbb{Z}, \\ (\max\limits_{0 \leq i < \eta} |a_i|) \cdot \eta(1 + \|\Gamma\|_\Delta)^{\eta-1} & \text{otherwise.} \end{cases}$$

and that for $a, b \in \mathbb{Z}[\theta]$, $\mathcal{H}(ab) \leq \mathcal{H}(a) \cdot \mathcal{H}(b)$ and $\mathcal{H}(a+b) \leq \mathcal{H}(a) + \mathcal{H}(b)$. Moreover, a can be represented as an integer polynomial of degree less than η with $O(\log \mathcal{H}(a))$ bits.

We represent an element $\alpha \in \mathbb{Q}(\theta)$ by $\alpha = a/b$, where $a \in \mathbb{Z}[\theta]$ as above and $b \in \mathbb{Z}$ is relatively prime to $\gcd(a_0, \ldots, a_{\eta-1})$. Define $\mathcal{H}(\alpha) = \max\{|b|, \mathcal{H}(a)\}$. It is easily verified that for $\alpha \in \mathbb{Q}$, $\mathcal{H}(1/\alpha) = \mathcal{H}(\alpha)$, while for general $\alpha \in \mathbb{Q}(\theta)$, $\mathcal{H}(1/\alpha) \leq \eta^\eta \mathcal{H}(a)^\eta$. We similarly extend the Δ-norm $\|\cdot\|_\Delta$ to matrices and polynomials over $\mathbb{Q}(\theta)$: for $B \in \mathbb{Q}(\theta)^{m \times n}$, $\|B\|_\Delta = \max_{ij} \mathcal{H}(B_{ij})$ and for $g = \sum_{0 \leq i \leq m} b_i x^i \in \mathbb{Q}(\theta)[x]$, $\|g\|_\Delta = \max_i \mathcal{H}(b_i)$.

Next suppose $p \in \mathbb{Z}$ is a prime such that $\Gamma \bmod p$ is irreducible in $\mathbb{F}_p[z]$. The prime p remains inert in the ring of integers $\mathcal{O}$ of $\mathbb{Q}(\theta)$, that is, the ideal $p\mathcal{O}$ is prime in $\mathcal{O}$. This also implies that $\mathbb{Z}[z]/(p, \Gamma) \cong \mathbb{F}_{p^\eta}$, the finite field with p^η elements. We can adjoin a root $\theta = (z \bmod \Gamma)$ of $\Gamma(z)$ to $\mathbb{Q}_p$ to obtain an extension field $\mathbb{Q}_p(\theta) \supseteq \mathbb{Q}_p$, called the localization of $\mathbb{Q}(\theta)$ at p. Similarly, we have $\mathbb{Z}_p[\theta]$, a ring extension of $\mathbb{Z}_p$ containing $\mathbb{Z}[\theta]$. $\mathbb{Z}_p[\theta]$ is easily shown to be a principal ideal domain (see Lang (1986), Section 2.1). It is also easily verified that $\mathbb{Z}_p[\theta]/(p, \Gamma) \cong \mathbb{F}_{p^\eta}$ (the *residue class field* of $\mathbb{Q}_p(\theta)$). Thus $[\mathbb{Z}_p[\theta] : \mathbb{Z}_p] = [\mathbb{Q}_p(\theta) : \mathbb{Q}_p] = \eta$ and $\{1, \theta, \theta^2, \ldots, \theta^{\eta-1}\}$ forms a $\mathbb{Z}_p$-basis for $\mathbb{Z}_p[\theta]$ and a $\mathbb{Q}_p$ basis for $\mathbb{Q}_p(\theta)$. We can extend the p-adic order and p-adic norm to $\mathbb{Q}_p(\theta)$ by letting $\mathrm{ord}_p(a) = \min\{\mathrm{ord}_p(a_i) : 0 \leq i < \eta\} \in \mathbb{Z}$ and $|a|_p = \max\{|a_i|_p : 0 \leq i < \eta\} \in \mathbb{R}_{\geq 0}$ for $a = \sum_{0 \leq i \leq \eta} a_i \theta^i \in \mathbb{Q}_p(\theta)$ (where $a_i \in \mathbb{Q}_p$). These definitions agree with the p-adic norm and order on $\mathbb{Q}_p$ on its embedding in $\mathbb{Q}_p(\theta)$. We then identify $\mathbb{Z}_p[\theta] = \{a \in \mathbb{Q}_p(\theta) : |a|_p \leq 1\}$.

In the language of p-adic analysis, $\mathbb{Q}_p(\theta)$ is the unique unramified extension field of degree η over $\mathbb{Q}_p$. $\mathbb{Z}_p[\theta]$ is the integral closure of $\mathbb{Z}_p$ in $\mathbb{Q}_p(\theta)$, that is, the elements of $\mathbb{Q}_p(\theta)$ which are roots of monic polynomials in $\mathbb{Z}_p[z]$. All this is in some sense made possible because p (or rather the principal ideal generated by p) remains prime in the ring of integers of $\mathbb{Q}(\theta)$. We obtain the following diagram of inclusions:



The utility in these definitions is in the following observation. Suppose we wish to evaluate a rational function $\Psi \in \mathbb{Z}(x_1, \ldots, x_n)$ (a quotient of integer polynomials) at a point $\bar{a} = (a_1, \ldots, a_n) \in \mathbb{Z}[\theta]^n$, say $b = \Psi(\bar{a}) \in \mathbb{Q}(\theta)$. Computationally b is represented by a polynomial $\sum_{0 \leq i < \eta} b_i z^i \in \mathbb{Q}[z]$. To show that a prime p does not divide any of the denominators of the b_i's, we can show that $b \in \mathbb{Z}_p[\theta]$. Since $\Psi \in \mathbb{Z}(x_1, \ldots, x_n) \subseteq \mathbb{Z}_p(x_1, \ldots, x_n)$ and $\bar{a} \in \mathbb{Z}^n \subseteq \mathbb{Z}_p^n$, we can view the computation as taking place over $\mathbb{Z}_p[\theta]$, which, unlike $\mathbb{Z}[\theta]$, is a PID. Obviously, this does not change the algorithm, only our perception of the space on which it operates.

5 Refining λ-smooth solutions to Diophantine solutions

We can now present our algorithm `RefineToDiophantine` to refine a $\lambda = 2r(r+1)$-smooth solution into a Diophantine solution. The algorithm is very similar to `SmoothSolver`, but works in a series of algebraic orders of very small degree over $\mathbb{Z}$.

Algorithm: `RefineToDiophantine`

Input: - $A \in \mathbb{Z}^{n\times n}$, $r = \operatorname{rank} A$ and $w \in \mathbb{Z}^{n\times 1}$;
- $v^{(0)} \in \mathbb{Q}^{n\times 1}$ such that $Av^{(0)} = w$ and $\delta_0 = \operatorname{denom}(v^{(0)})$ is $2r(r+1)$-smooth;
- an error tolerance $\epsilon > 0$;

Output: - $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ or a report "No Integer Solution Exists";

(1) Let $p_1, \ldots, p_\kappa \le 2r(r+1)$ be the distinct primes dividing δ_0;
(2) Using `BuildOrders` on inputs $\eta = \lceil \log_2(2r(r+1)) \rceil$ and $p_1, \ldots, p_\kappa$, find a set $G = \{\Gamma_1, \ldots, \Gamma_l\} \subseteq \mathbb{Z}[z]$ of monic polynomials of degree η such that for each p_i there exists a $\Gamma_j \in G$ such that $\Gamma_j \bmod p_i$ is irreducible in $\mathbb{F}_p[z]$; Let $\theta_j = (z \bmod \Gamma_j)$;
(3) Let $g := \delta_0$;
(4) For $c := 1$ to $\lceil 1 + \log_2(1/\epsilon) \rceil$ While $g \neq 1$ Do
(5) For $i := 1$ to $s := \lceil 1 + \log_2(\kappa) \rceil$ Do
(6) For $j := 1$ to l Do
(7) Let $\mathcal{V}_j = \{\sum_{0\le k<\eta} a_k \theta_j^k : a_k \in \{0,1\}\} \subseteq \mathbb{Z}[\theta_j]$;
(8) Choose random $u_2, \ldots, u_n, l_2, \ldots, l_n \in \mathcal{V}_j$;
"Build" a black box for $B = UAL$ with U, L as in (2.1); Let $B_r = B\begin{bmatrix}1\ldots r\\ 1\ldots r\end{bmatrix}$;
Let $\bar{w} := Uw = (\bar{w}_1, \ldots, \bar{w}_n)^t \in \mathbb{Z}[\theta_j]^{n\times 1}$;
(9) Solve $B_r \bar{v} = (\bar{w}_1, \ldots, \bar{w}_r)^t$ for $\bar{v} = (\bar{v}_1, \ldots, \bar{v}_r)^t \in \mathbb{Q}(\theta_j)^{r\times 1}$;
If B_r is singular, goto (8);
(10) Let $\sum_{0\le k<\eta} v_k^{(i,j)} \theta^k := L(\bar{v}_1, \ldots, \bar{v}_r, 0, \ldots, 0)^t$, where $v_k^{(i,j)} \in \mathbb{Q}^{n\times 1}$ for $0 \le k < \eta$;
Let $v^{(i,j)} := v_0^{(i,j)} \in \mathbb{Q}^{n\times 1}$ and $\delta_{i,j} := \operatorname{denom}(v^{(i,j)})$;
If $Av^{(i,j)} \neq w$ then report "No solution to Diophantine system exists";
(11) Let $g := \gcd(g, \delta_{i,j})$;
End For;
End For;
End For;
If $g = 1$ Then
(12) Find $\gamma_0, \gamma_{i,j} \in \mathbb{Z}$ for $1 \le i \le s$ and $1 \le j \le l$ such that $\gamma_0\delta_0 + \sum \gamma_{i,j}\delta_{i,j} = 1$;
(13) Return $v := \gamma_0\delta_0 v^{(0)} + \sum \gamma_{i,j}\delta_{i,j} v^{(i,j)} \in \mathbb{Z}^{n\times 1}$;
Else Report "No solution to Diophantine system exists".
End If;

THEOREM 5.1. *The algorithm* `RefineToDiophantine` *works as specified.*

(i) If a solution $v \in \mathbb{Z}^{n\times 1}$ is returned, it is always correct;
(ii) $\log \|v\|_\Delta = \tilde{O}(r \log n + r \log \|A\|_\Delta + \log \|w\|_\Delta)$ when $\log \|v^{(0)}\|_\Delta$ is of this same size;
(iii) If an integer solution exists to the system, a solution is found with probability at least $1 - \epsilon$.

PROOF. The proof follows in much the same way as Theorem 3.1. For part (i), we note that $A(\delta_{i,j}v^{(i,j)}) = \delta_{i,j}w$ for $1 \le i \le s$ and $1 \le j \le l$. Thus

$$Av = A\left(\sum_{\substack{1\le i\le s\\ 1\le j\le l}} \gamma_{i,j}\delta_{i,j}\cdot v^{(i,j)}\right) = \left(\sum_{\substack{1\le i\le s\\ 1\le j\le l}} \gamma_{i,j}\delta_{i,j}\right)\cdot w = w$$

and $v \in \mathbb{Z}^{n\times 1}$

For parts (ii) and (iii), first consider an iteration of the inner loop (7)-(11). We have

$$\begin{aligned}\|B\|_\Delta &\le n^2\|U\|_\Delta\cdot\|A\|_\Delta\cdot\|L\|_\Delta = O(n^2\cdot\|A\|_\Delta\cdot(\eta(1+\|\Gamma\|_\Delta)^{\eta-1})^2)\\ \|Uw\|_\Delta &\le n\|U\|_\Delta\cdot\|w\|_\Delta = O(n\|w\|_\Delta\cdot\eta(1+\|\Gamma\|_\Delta)^{\eta-1}).\end{aligned}$$

Applying Hadamard's bound and Cramer's rule we find

$$\begin{aligned}\log\|\det(B_r)\|_\Delta &= O(r\log n + r\log\|A\|_\Delta + r\eta\log\|\Gamma\|_\Delta),\\ \log|\delta_{i,j}| \le \log\|1/\det(B_r)\|_\Delta &= O(r\eta\log n + r\eta\log\|A\|_\Delta + r\eta^2\log\|\Gamma\|_\Delta),\\ \log\|\det(B_r)\bar{v}\|_\Delta &= O(r\log n + r\log\|A\|_\Delta + \log\|w\|_\Delta + r\eta\log\|\Gamma\|_\Delta),\\ \log\|v^{(i,j)}\|_\Delta &= O(r\eta\log n + r\eta\log\|A\|_\Delta + \log\|w\|_\Delta + r\eta^2\log\|\Gamma\|_\Delta),\end{aligned}$$

for $1 \le i \le s$ and $1 \le j \le l$.

Since $B\binom{1\ldots r}{1\ldots r}$ is a non-zero polynomial in $u_2,\ldots,u_n$, $l_2,\ldots,l_n$ of degree $2r$, B_r is non-singular with probability at least $1-2r/(2r(r+1)) = r/(r+1)$ by Fact 2.4. Thus we expect to execute steps (8) and (9) a constant number of times for each iteration of the inner For loop. If a solution to $Av = w$ exists over $\mathbb{Z}$ it certainly exists over $\mathbb{Q}(\theta_j)$, and by Lemma 2.6 $v^{(i,j)}$ will be such a solution (since $\mathbb{Q}(\theta_j)$ is a field and PID). Once the GCD of the denominators is one, we execute steps (12) and (13), as in `SmoothSolver`. Iliopolous's (1989) algorithm finds $\gamma_{ij} \in \mathbb{Z}$ such that $\log|\gamma_{i,j}| = O(\log\max_{i,j}|\delta_{i,j}|\cdot\log(sl))$. The constructed v satisfies

$$\begin{aligned}\log\|v\|_\Delta &= \log\left(\sum_{\substack{1\le i\le s\\ 1\le j\le l}}\gamma_{i,j}\|\delta_{i,j}v^{(i,j)}\|_\Delta\right)\\ &= O((r\eta\log n + r\eta\log\|A\|_\Delta + r\eta^2\log\|\Gamma\|_\Delta)\cdot\log(rl) + \log\|w\|_\Delta).\end{aligned}$$

Since $\eta = O(\log r)$, $l = O(\log^2 r)$ and $\log\|\Gamma\|_\Delta = O(\log^2 r)$ by Theorem 4.1, $\log\|v\|_\Delta = O(r\log n + r\log\|A\|_\Delta + \log\|w\|_\Delta)$ which proves (ii).

To prove (iii) assume that a Diophantine solution does indeed exist. We show that with each iteration of the outer For loop, the algorithm finds such a solution with probability at least $1/2$. Let $p \in \{p_1,\ldots,p_\kappa\}$ and suppose that $\Gamma_j \in G$ is irreducible modulo p and $\theta_j = (z \bmod \Gamma_j)$. Let B_p be the image of B in $\mathbb{Z}_p(\theta_j)^{r\times r}$. Since $\#(\mathcal{V}_j \bmod p) \ge 2r(r+1)$, by Theorem 2.5,

$$\text{Prob}\left\{\text{ord}_p B_p\binom{1\ldots k}{1\ldots k} = \text{ord}_p d_k \quad \forall k : 1 \le k \le r\right\} \ge 1/2.$$

If indeed $\operatorname{ord}_p B_p\binom{1\ldots k}{1\ldots k} = \operatorname{ord}_p d_k$ for all k $(1 \le k \le r)$, B_p is Smith dominant. Thus by Theorem 2.1, the image of $\bar{v}$ in $\mathbb{Q}_p(\theta_j)^{r\times 1}$ lies in $\mathbb{Z}_p[\theta_j]^{r\times 1}$, and the image of $v^{(i,j)}$ in $\mathbb{Q}_p(\theta_j)^{n\times 1}$ lies in $\mathbb{Z}_p[\theta_j]^{n\times 1}$, whence $p \nmid \operatorname{denom}(v^{(i,j)})$. Since $A \in \mathbb{Z}^{n\times n}$ and $w \in \mathbb{Z}^{n\times 1}$, $Av^{(i,j)} = Av_0^{(i,j)} = w$ and $Av_k^{(i,j)} = 0$ for $1 \le k < \eta$. Thus, the probability that $p \mid \operatorname{denom}(v^{(i,j)})$ for all $1 \le i \le s$ is at most $1/(2\kappa)$ and the probability this is true for any prime $p \in \{p_1, \ldots, p_\kappa\}$ is thus at most $1/2$. By executing the outer For loop $\lceil 1 + \log_2(1/\epsilon)\rceil$ times we ensure that if a solution exists, we will find one with probability at least $1 - \epsilon$. □

Like `SmoothSolver`, `RefineToDiophantine` can be applied to rational matrices as a direct application of the techniques of Theorem 2.8. We first examine the cost of solving non-singular systems over a number field using the Block-Wiedemann algorithm.

THEOREM 5.2. *Let $\Gamma \in \mathbb{Z}[z]$ be irreducible of degree $\eta = O(\log r)$ with $\log \mathcal{H}(\Gamma) = O(\log^2 r)$ and $\theta = (z \bmod \Gamma)$. Suppose we are given a black box for a non-singular matrix $B \in \mathbb{Z}[\theta]^{r\times r}$ and vector $\bar{w} \in \mathbb{Z}[\theta]^{r\times 1}$ and wish to solve $B\bar{v} = \bar{w}$ for $\bar{v} \in \mathbb{Q}(\theta)^{r\times 1}$. Let $\varrho = r\log\|B\|_\Delta + \log\|\bar{w}\|_\Delta$. On a network of $N \le r\varrho$ processors we can solve for $\bar{v}$ with an expected $O^\sim(r\varrho/N)$ matrix-vector products by B modulo (single word) primes with $O(\log r + \log\log(\|B\|_\Delta + \|\bar{w}\|_\Delta))$ bits. An additional $O^\sim(r^2 + r\,\mathrm{M}(\varrho)/\min(r,N))$ bit operations is executed simultaneously by each processor. Each processor requires additional storage for $O^\sim(r\rho/\min(r,N))$ words (not including possibly shared images of B modulo single-word primes).*

PROOF. We will apply Theorem 2.8 in a somewhat more complicated way than in Theorem 3.3. The set $\mathcal{B}$ of bad primes will consist of those primes p which either (i) divide the discriminant of Γ (so $\Gamma \bmod p$ is not squarefree; there are $O(\eta\log\eta + \eta\log\|\Gamma\|_\Delta)$ such primes), (ii) are such that $(\det B \bmod p)$ is not a unit in the finite ring $\mathbb{Z}[z]/(\Gamma, p)$ (so B is not invertible modulo p; there are $O(\eta\log\eta + \eta r\log r + \eta r\log\|B\|_\Delta)$ of these), or (iii) are less than $16r^2$ (there are $O(r^2/\log(r))$ of these). Thus $\#\mathcal{B}$ is polynomial in the logarithm of the output height $O(r\log r + r\log\|B\|_\Delta)$.

After constructing a set of small primes $\mathcal{P}$ as in Theorem 2.8 (immediately eliminating those bad primes falling into cases (i) and (iii) above), the computation proceeds by completely factoring $(\Gamma \bmod p) \equiv \Gamma_1^{(p)} \cdots \Gamma_k^{(p)}$, where $\Gamma_i \in \mathbb{F}_p[z]$. We then apply the block-Wiedemann algorithm over the finite fields $\Gamma[z]/(\Gamma_i^{(p)}, p)$ for $1 \le i \le k$ (see Fact 3.2). The solution is first recovered by the Chinese remainder algorithm to get a solution in $(\mathbb{F}_p[z]/(\Gamma, p))^{r\times 1}$ and finally a solution in $\mathbb{Q}(\theta)^{r\times 1}$.

The execution costs can now be estimated as in Theorem 3.3. □

THEOREM 5.3. *The algorithm* `RefineToDiophantine` *works as stated on input $A \in \mathbb{Z}^{n\times n}$ with rank r, $w \in \mathbb{Z}^{n\times 1}$, $v^{(0)} \in \mathbb{Q}^{n\times 1}$ with $\delta_0 = \operatorname{denom}(v^{(0)})$ being $2r(r+1)$-smooth, and $\epsilon > 0$. Let $\varrho = r\log\|A\|_\Delta + \log\|w\|_\Delta$ and suppose we are computing on a network of $N \le r\varrho$ processors.*

(i) If a Diophantine solution $v \in \mathbb{Z}^{n\times 1}$ to $Av = w$ exists, `RefineToDiophantine` *finds one with an expected number of $O^\sim(r\varrho/N)$ matrix-vector products by A modulo primes with $O(\log n + \log\log(\|A\|_\Delta + \|w\|_\Delta))$ bits. An additional $O^\sim(r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))$ bit operations is executed simultaneously by each processor.*

(ii) If no Diophantine solution exists, `RefineToDiophantine` *requires an expected number of* $O^{\sim}((r\varrho/N)\cdot\log(1/\epsilon))$ *matrix-vector products by* A *modulo primes with* $O(\log n + \log\log(\|A\|_\Delta + \|w\|_\Delta))$ *bits. An additional* $O^{\sim}((r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))\cdot\log(1/\epsilon))$ *bit operations is executed simultaneously by each processor.*

Each processor requires additional storage for $O^{\sim}(n + n\rho/\min(n,N))$ *words (not including possibly shared images of* A *modulo single-word primes).*

PROOF. The number of elements in G governs the number l of iterations of the innermost For loop; by Theorem 4.1 $l = O(\log^2 r)$. Thus we execute (7)-(11) an expected number $2ls = O(\log^3 r)$ times if a solution exists and $O(\log^3(r)\cdot\log(1/\epsilon))$ times otherwise.

Each evaluation of the black box for $y \to B_r y$ where $y = (y_1,\ldots,y_r)^t \in \mathbb{Z}[\theta_j]^{r\times 1}$ is performed by evaluating $UAL(y_1,\ldots,y_r,0,\ldots,0)^t = (z_1,\ldots,z_n)^t$, and returning $(z_1,\ldots,z_r)^t = B_r y$. Thus each matrix-vector product by B_r requires one black box evaluation of A modulo primes with $O(\log r + \log\log(\|B\|_\Delta + \|\bar{w}\|_\Delta))$ or $O(\log n + \log\log(\|A\|_\Delta + \|w\|_\Delta))$ bits, plus $O(n\log n\cdot\eta\log\eta)$ additional operations in $\mathbb{Z}[z]/(p,\Gamma)$ for primes p of this same size. The linear system $B_r\bar{v} = \bar{w}$ in step (6) is then solved using the Block-Wiedemann method described in Theorem 5.2. The remaining cost analysis follows in the same manner as Theorem 3.4. □

Given $A \in \mathbb{Z}^{n\times n}$ and $w \in \mathbb{Z}^{n\times 1}$, a complete algorithm for finding a Diophantine solution $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ is obtained by first applying `SmoothSolver` with smoothness bound $\lambda = 2r(r+1)$ (to get an solution with λ-smooth denominator) followed by `RefineToDiophantine` (using the λ-smooth solution as additional input $v^{(0)}$). We immediately obtain the following corollary.

COROLLARY 5.4. *Let* $A \in \mathbb{Z}^{n\times n}$ *with rank* r, $w \in \mathbb{Z}^{n\times 1}$ *and* $\epsilon > 0$. *Let* $\varrho = r\log\|A\|_\Delta + \log\|w\|_\Delta$ *and suppose we are computing on a network of* $N \le r\varrho$ *processors.*

(i) If a Diophantine solution $v \in \mathbb{Z}^{n\times 1}$ *to* $Av = w$ *exists, we can find one with an expected number of* $O^{\sim}(r\varrho/N)$ *matrix-vector products by* A *modulo primes with* $O(\log n + \log\log(\|A\|_\Delta + \|w\|_\Delta))$ *bits. An additional* $O^{\sim}(r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))$ *bit operations is executed simultaneously by each processor. The returned* $v \in \mathbb{Z}^{n\times 1}$ *satisfies* $\log\|v\|_\Delta = O^{\sim}(r\log n + r\log\|A\|_\Delta + \log\|w\|_\Delta)$.

(ii) If no Diophantine solution exists, we can determine this with an expected number of $O^{\sim}((r\varrho/N)\cdot\log(1/\epsilon))$ *matrix-vector products by* A *modulo primes with* $O(\log n + \log\log(\|A\|_\Delta + \|w\|_\Delta))$ *bits; an additional* $O^{\sim}((r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))\cdot\log(1/\epsilon))$ *bit operations is executed simultaneously by each processor.*

An incorrect solution is never returned. If any solution exists, one is found with probability at least $1-\epsilon$. *Each processor requires additional storage for* $O^{\sim}(n + n\rho/\min(n,N))$ *words (not including possibly shared images of* A *modulo single-word primes).*

6 Open Questions

A number of important questions remain unresolved and extensions remain unexplored.

Random generation of Diophantine solutions. While the solutions we generate are in some sense random, it has not been proven that they in any way sample uniformly

from the solution space. Clearly, it is not possible to write down a complete basis for the solution space within the amount of time and space allowed. Still, Kaltofen & Saunders (1991) showed how to randomly sample from the solution manifold for singular systems of linear equations over a field. Such a result should be obtainable in the current context.

Proving SmoothSolver yields Diophantine solutions directly. `SmoothSolver` is currently only shown to give solutions whose denominators are $2r(r+1)$-smooth. These are later refined to integer solutions by `RefineToDiophantine`. It seems quite possible that `SmoothSolver` finds Diophantine solutions quickly as well, but this appears difficult to prove. The problem seems akin to showing Coppersmith's (1994) algorithm works over $\mathbb{F}_2$; see Kaltofen (1995).

Finding positive solutions It is desirable in many applications (such as combinatorics) to obtain solutions whose coefficients are all positive. Ad hoc techniques based on combining multiple general solutions have been used in some cases for finding certain combinatorial designs. It would be useful to find a rigorous method based on a small number of random solutions.

Implementation The algorithms discussed here are currently being implemented using the LiDIA library for computational number theory.

References

E. Bach and J. Shallit. *Algorithmic Number Theory, Volume 1: Efficient Algorithms.* MIT Press (Cambridge, MA), 1996.

E. R. Berlekamp. Factoring polynomials over large finite fields. *Math. Comp.* **24**, pp. 713–735, 1970.

W. A. Blankinship. Algorithm 288, solution of simultaneous linear diophantine equations. *Comm. ACM* **9**, pp. 514, 1966.

I. Borosh and A. S. Fraenkel. Exact solutions of linear equations with rational coefficients by congruence techniques. *Mathematics of Computation* **20**, pp. 107–112, 1966.

G. Bradley. Algorithms for Hermite and Smith normal matrices and linear diophantine equations. *Math. Comp* **25**(116), pp. 897–907, 1971.

J.W.S. Cassels. *Local Fields*, vol. 3 of *London Mathematical Society Student Texts.* Cambridge University Press, 1986.

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.

H. Cohen. *A Course in Computational Number Theory.* Springer, 1993.

G. Collins and M. Encarnación. Efficient rational number reconstructions. *Journal of Symbolic Computation* **20**, pp. 287–297, 1995.

D. Coppersmith. Solving homogeneous linear equations over $gf(2)$ via block wiedemann algorithm. *Mathematics of Computation* **62**(205), pp. 333–350, 1994.

M. A. Frumkin. An application of modular arithmetic to the construction of algorithms for solving systems of linear equations. *Dokl. Akad. Nauk SSSR* **17**(4), pp. 1165–1168, 1976.

F. R. Gantmacher. *The Theory of Matrices, Vol. I.* Chelsea Publishing Co. (New York NY), 1990.

M. Giesbrecht. Nearly optimal algorithms for canonical matrix forms. *SIAM J. Comp.* **24**, pp. 948–969, 1995.

M. Giesbrecht. Fast computation of the smith form of a sparse integer matrix. *Computational Complexity* , 1996. Submitted.

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.

G. Havas and B.S. Majewski. Hermite normal form computation for integer matrices. *Congressus Numerantium* **105**, pp. 184–193, 1994.

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.

T. Kailath. *Linear systems.* Prentice-Hall (Englewood Cliffs, New Jersey), 1980.

E. Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Mathematics of Computation* **64**(210), pp. 777–806, 1995.

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.

E.S. Kramer and D. Mesner. t-designs on hypergraphs. *Discrete Math.* **15**, pp. 262–296, 1976.

S. Lang. *Algebraic Number Theory.* Springer-Verlag (New York), 1986.

R. Lidl and H. Niederreiter. *Finite Fields*, vol. 20 of *Encyclopedia of Mathematics and its Applications.* Addison-Wesley (Reading MA), 1983.

B. Majewski and G. Havas. A solution to the extended gcd problem. In *Proc. ISSAC'95*, pp. 248–253, Montreal, Canada, 1995.

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.

A. Storjohann. A fast practical deterministic algorithm for triangularizing integer matrices. Unpublished manuscript, 1996.

A. Storjohann and G. Labahn. Asymptotically fast computation of Hermite normal forms of integer matrices. In *Proceedings of ISSAC'96*, pp. 259–266, Zurich, Switzerland, 1996.

P. Wang, M. Guy, and J. Davenport. *P*-adic reconstruction of rational numbers. *SIGSAM Bulletin* **16**(2), pp. 2–3, 1982.

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

R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proc. EUROSAM 79*, pp. 216–226, Marseille, 1979.